Figure 1

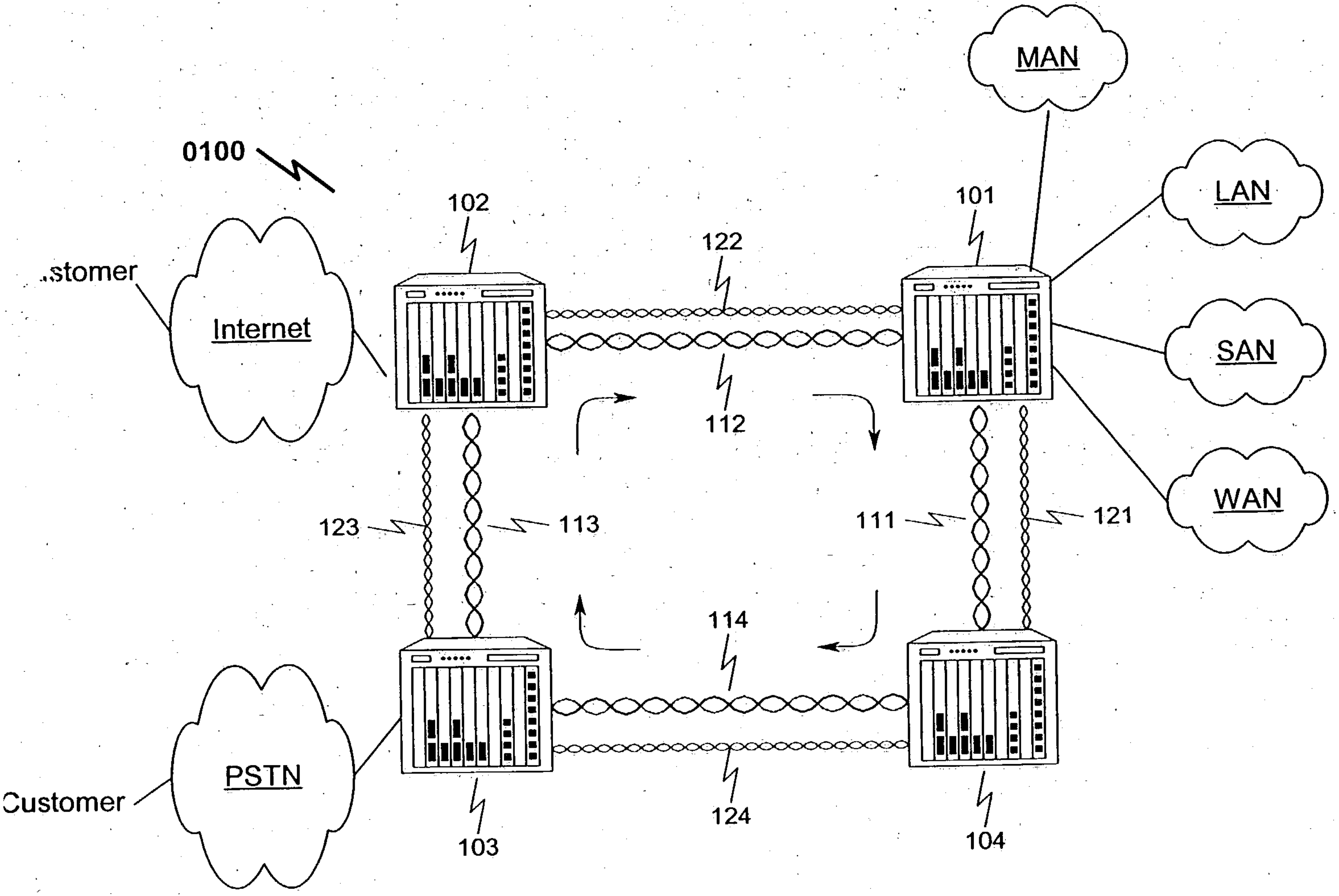
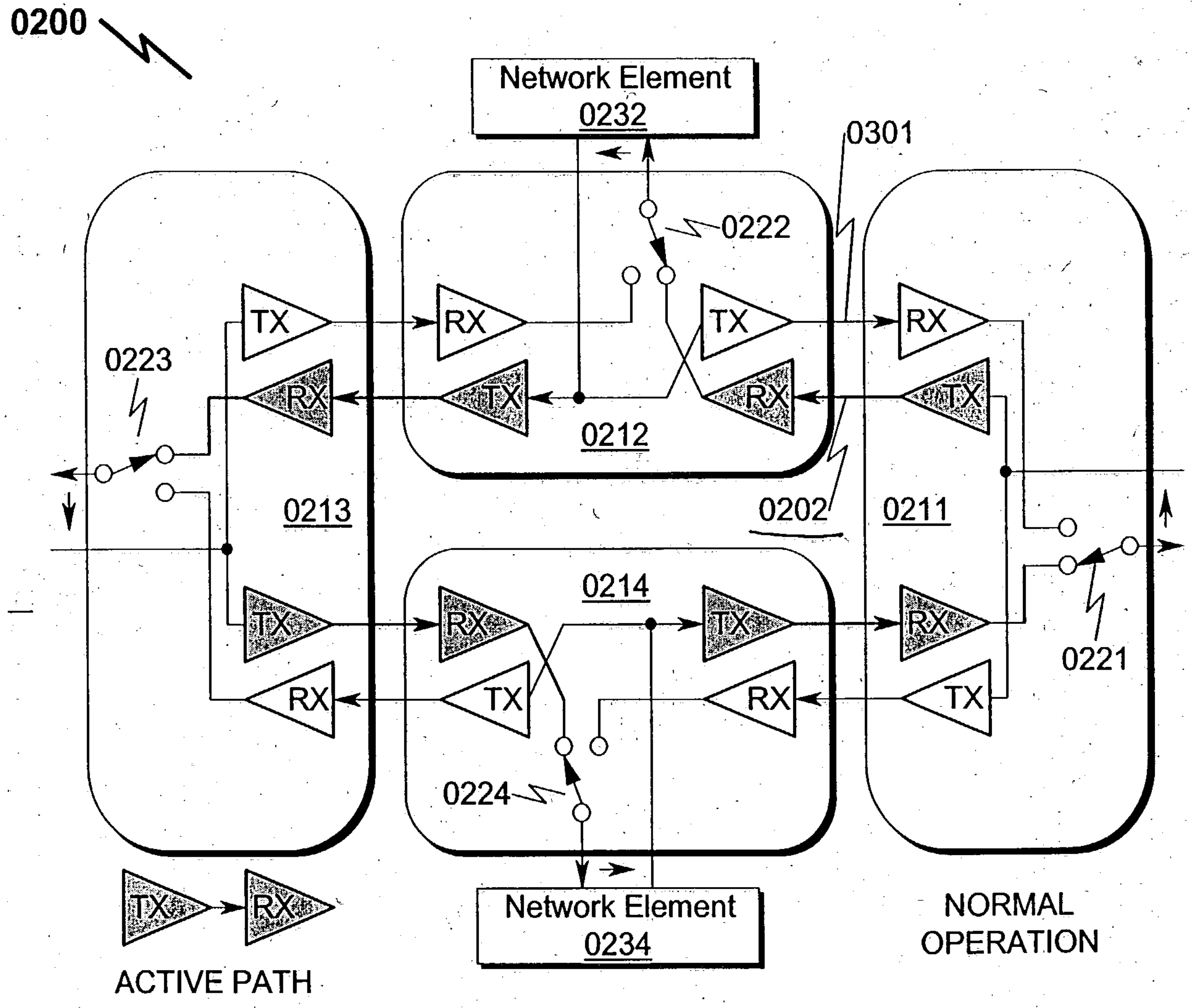

MAN
0100
LAN
102
101
122
.stomer
Internet
SAN
112
WAN
123
113
111
121
114
PSTN
Customer
103
124
104

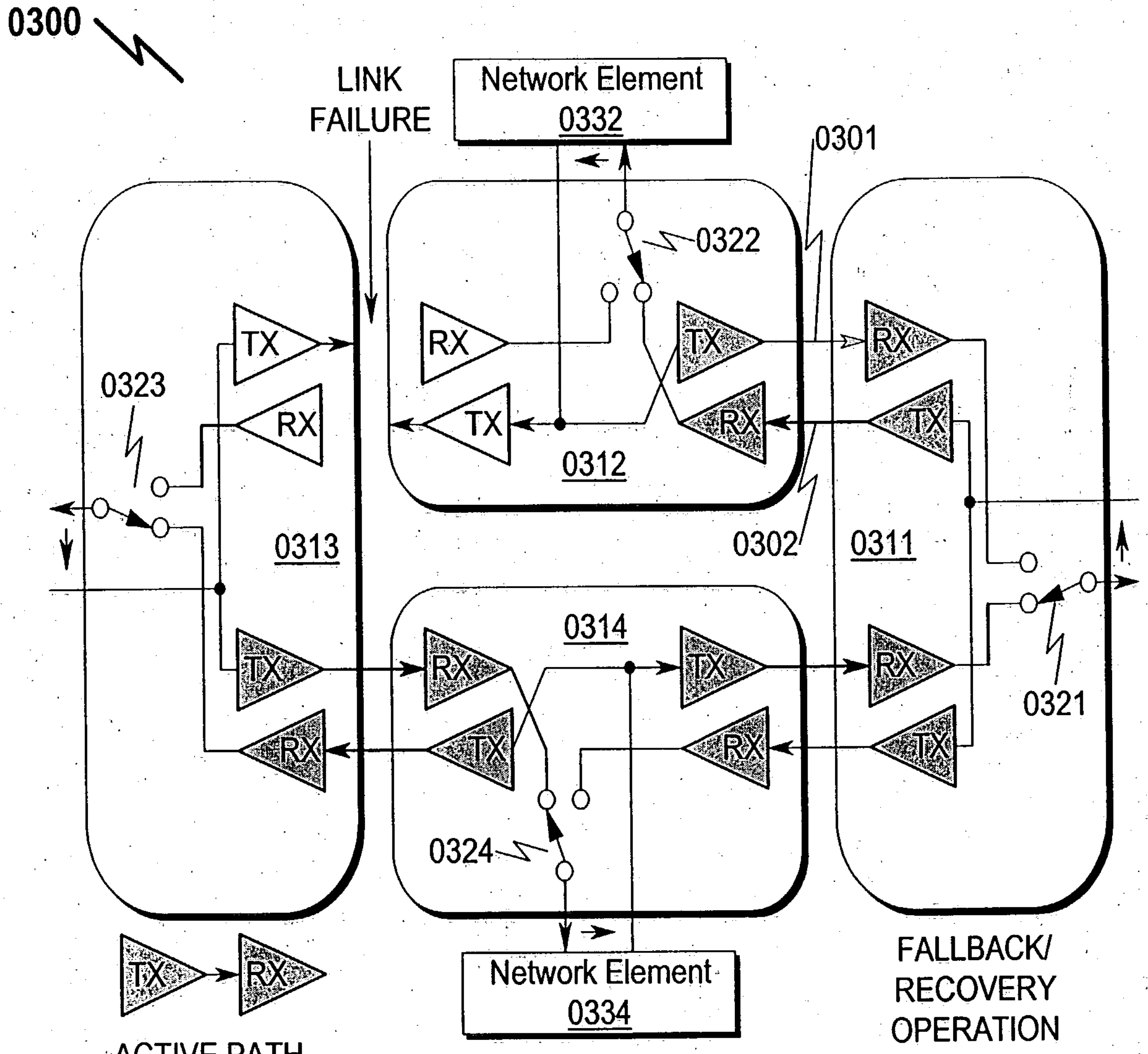
Figure 2



PRIOR ART

2017.

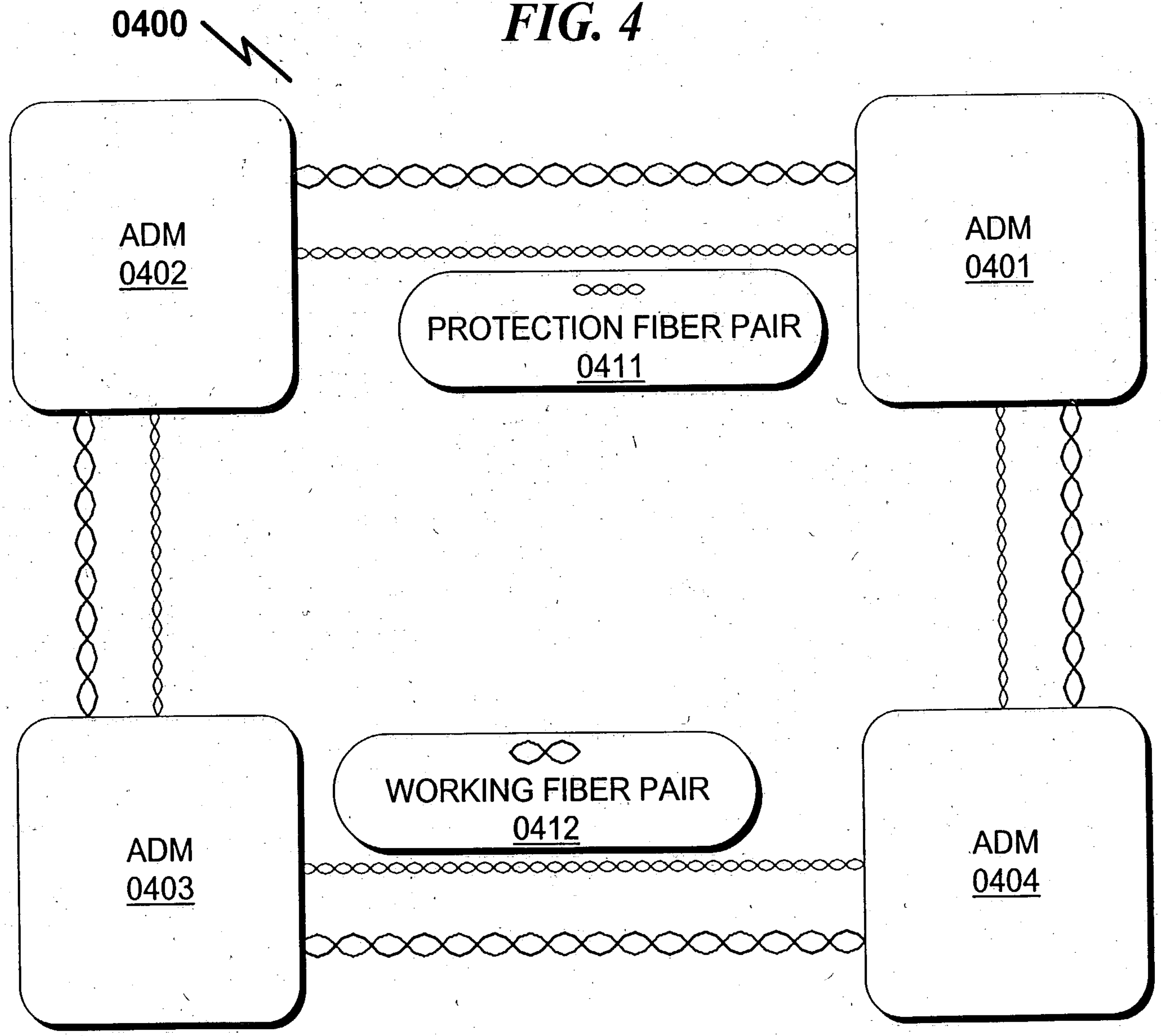
Figure 3



PRIOR ART

Figure 4

FIG. 4



PRIOR ART

Figure 5

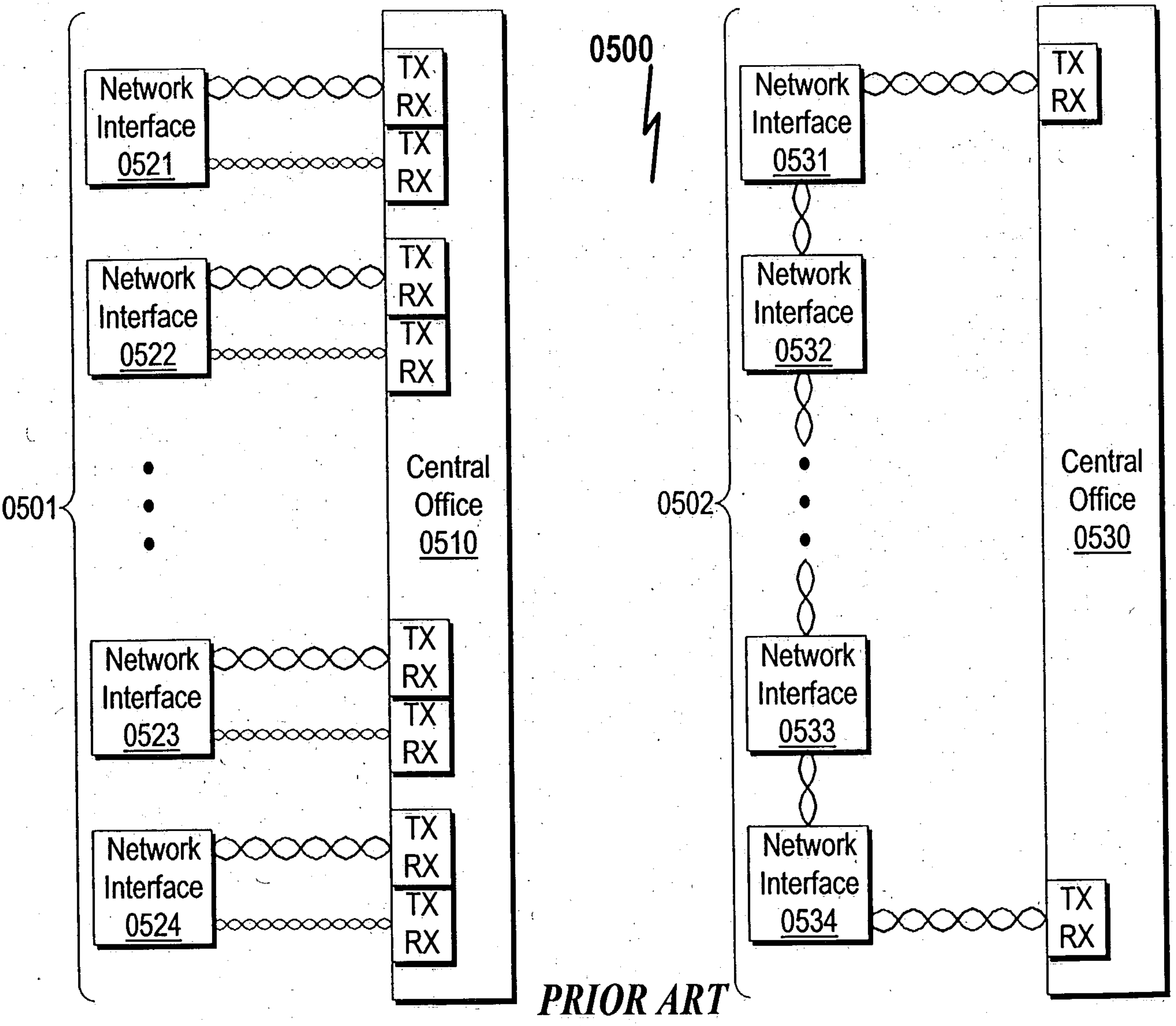
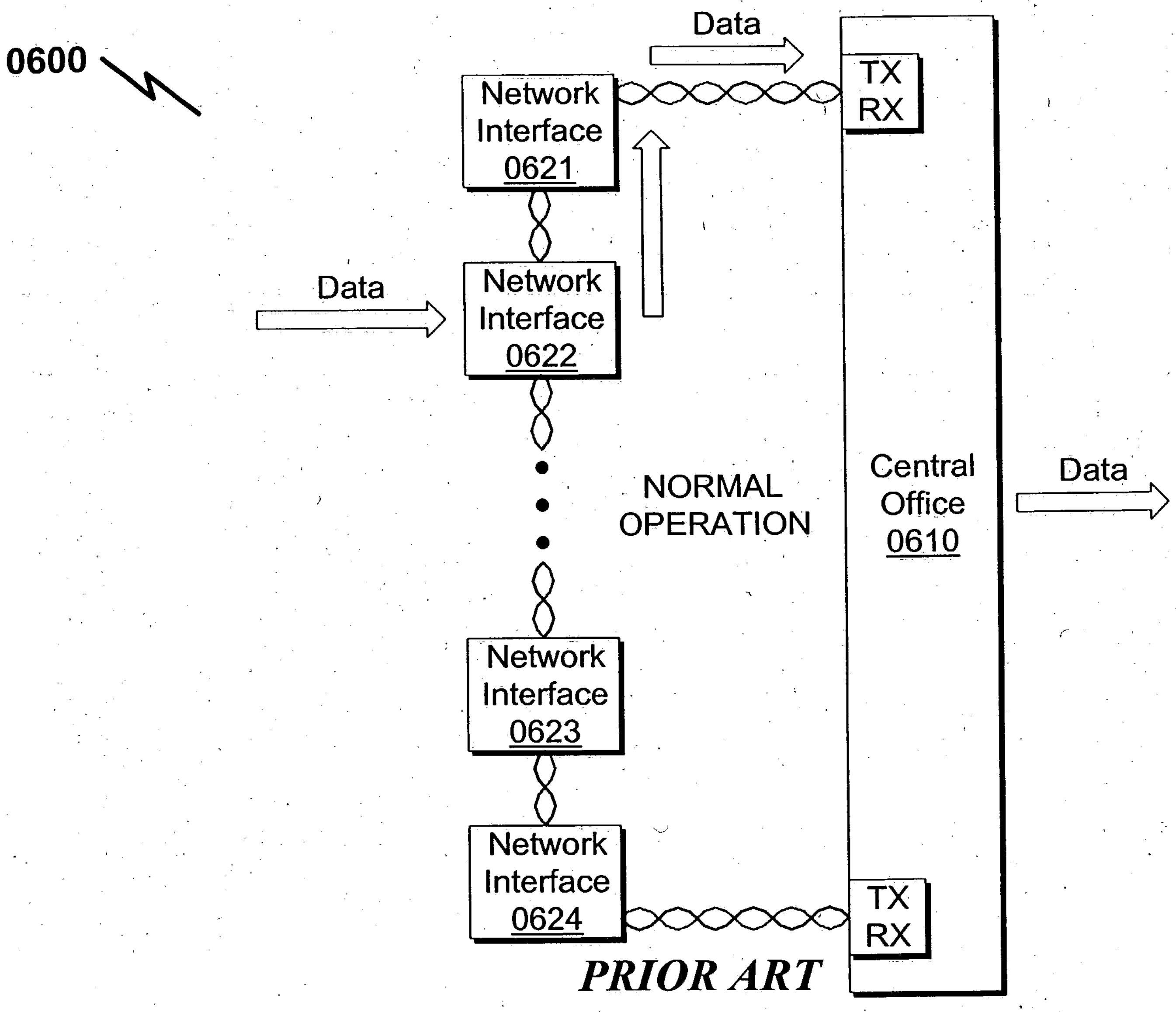

0500
0501
Network Interface 0521
Network Interface 0522
Network Interface 0523
Network Interface 0524
TX
RX
Central Office 0510
0502
Network Interface 0531
Network Interface 0532
Network Interface 0533
Network Interface 0534
Central Office 0530
PRIOR ART

Figure 6



PRIOR ART

Figure 7

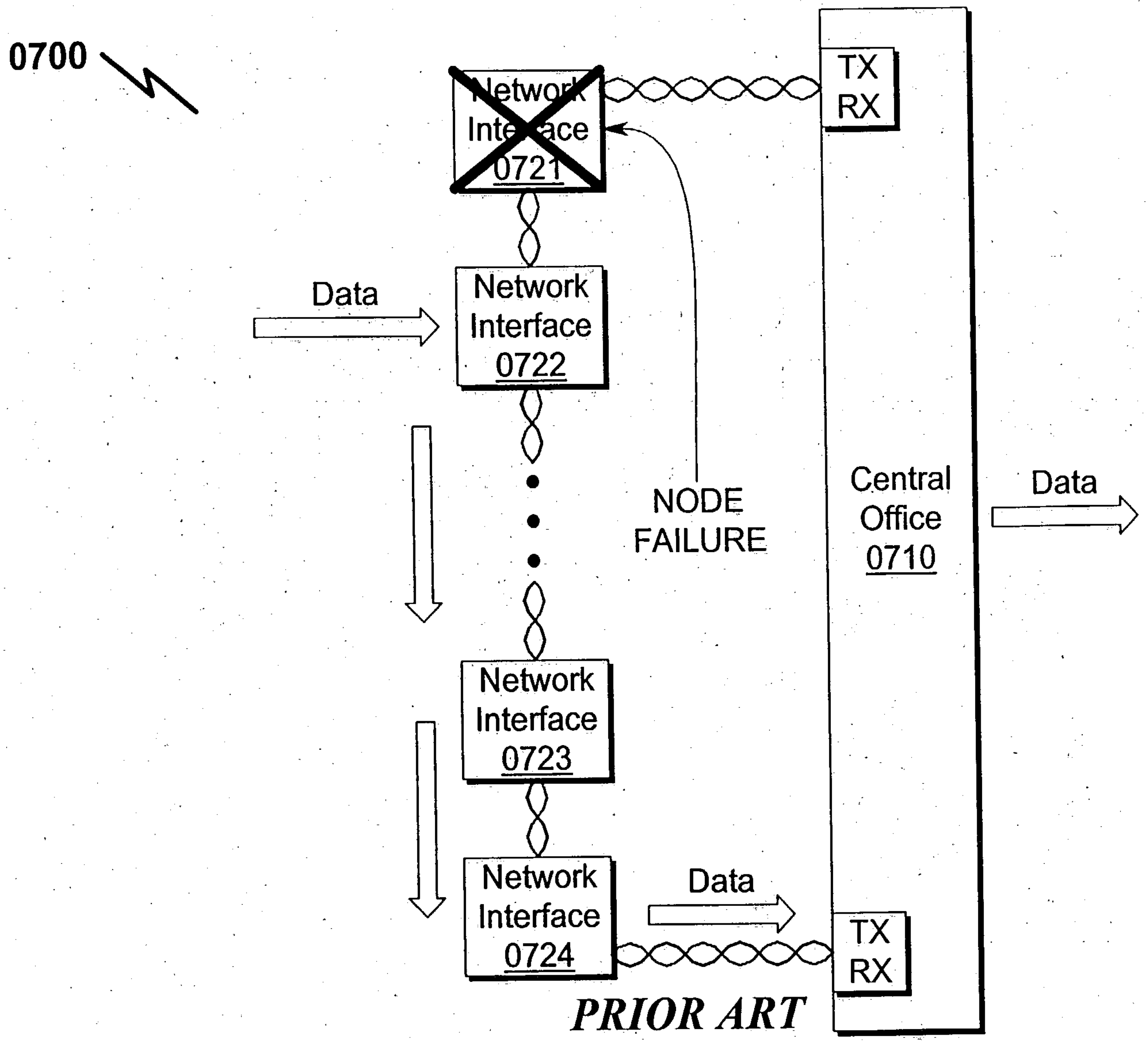
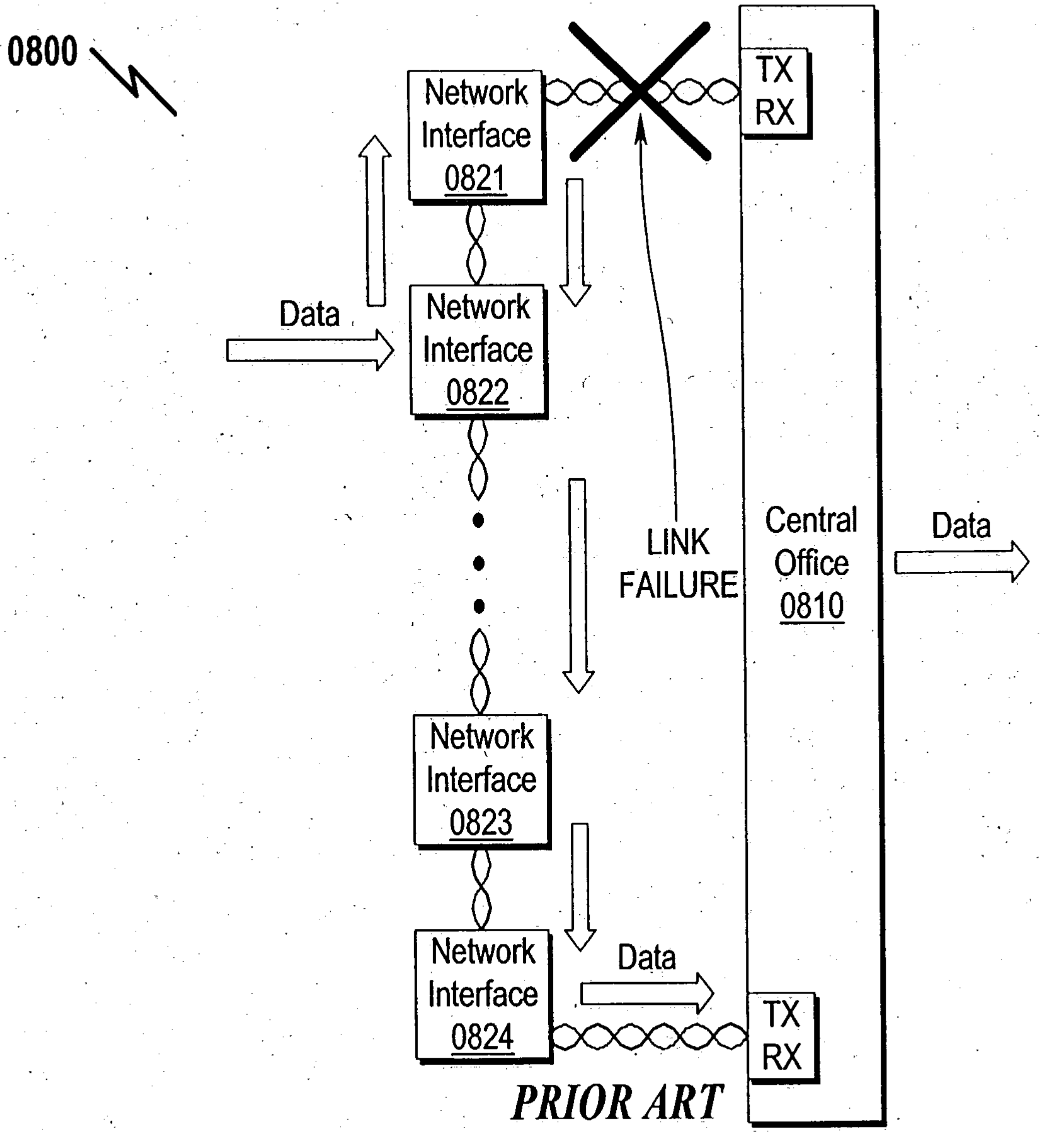
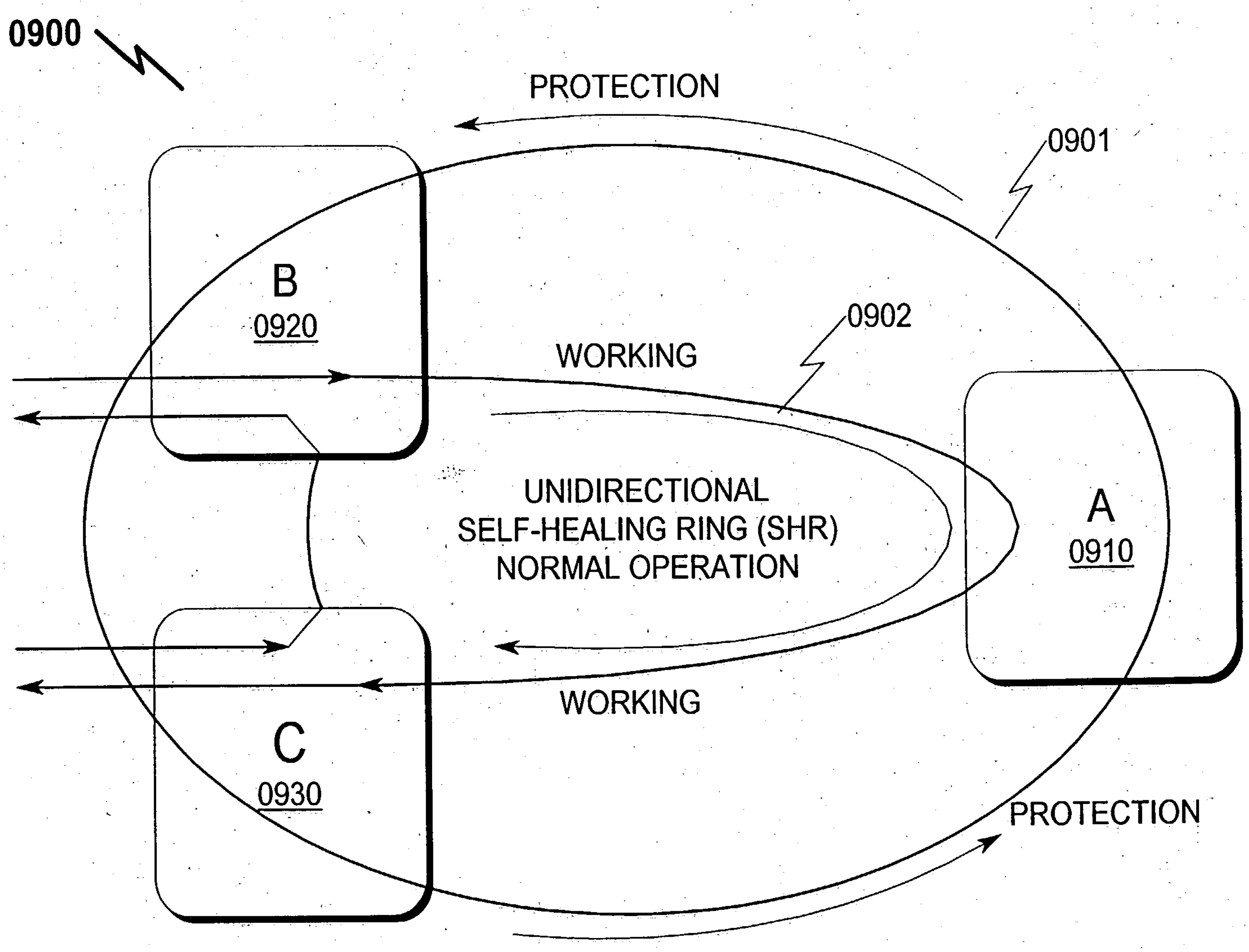

0700
Network Interface 0721
TX
RX
Data
Network Interface 0722
NODE FAILURE
Central Office 0710
Data
Network Interface 0723
Network Interface 0724
Data
TX
RX
PRIOR ART

Figure 8



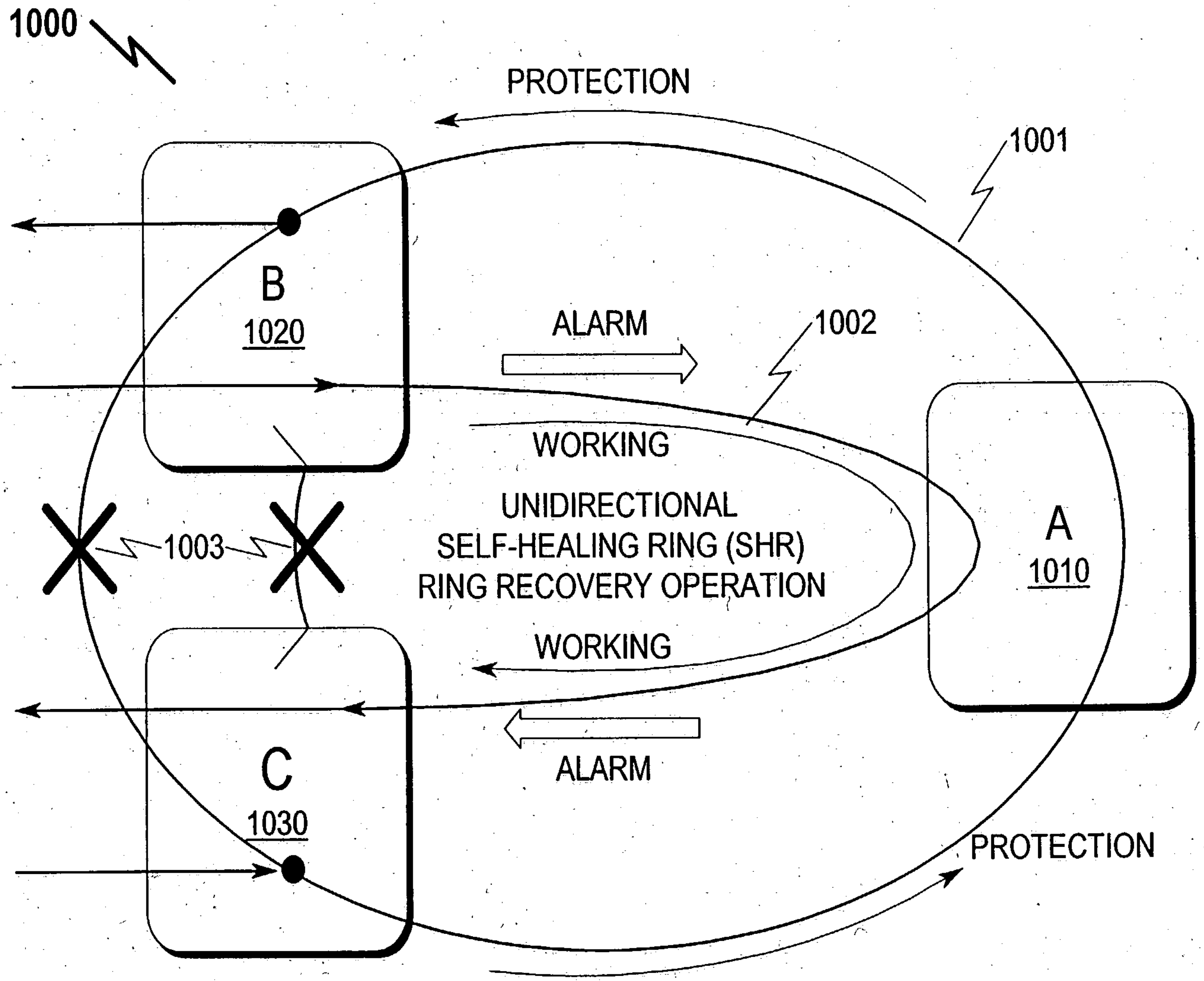
PRIOR ART

Figure 9



PRIOR ART

Figure 10



PRIOR ART

Figure 11
Prior Art

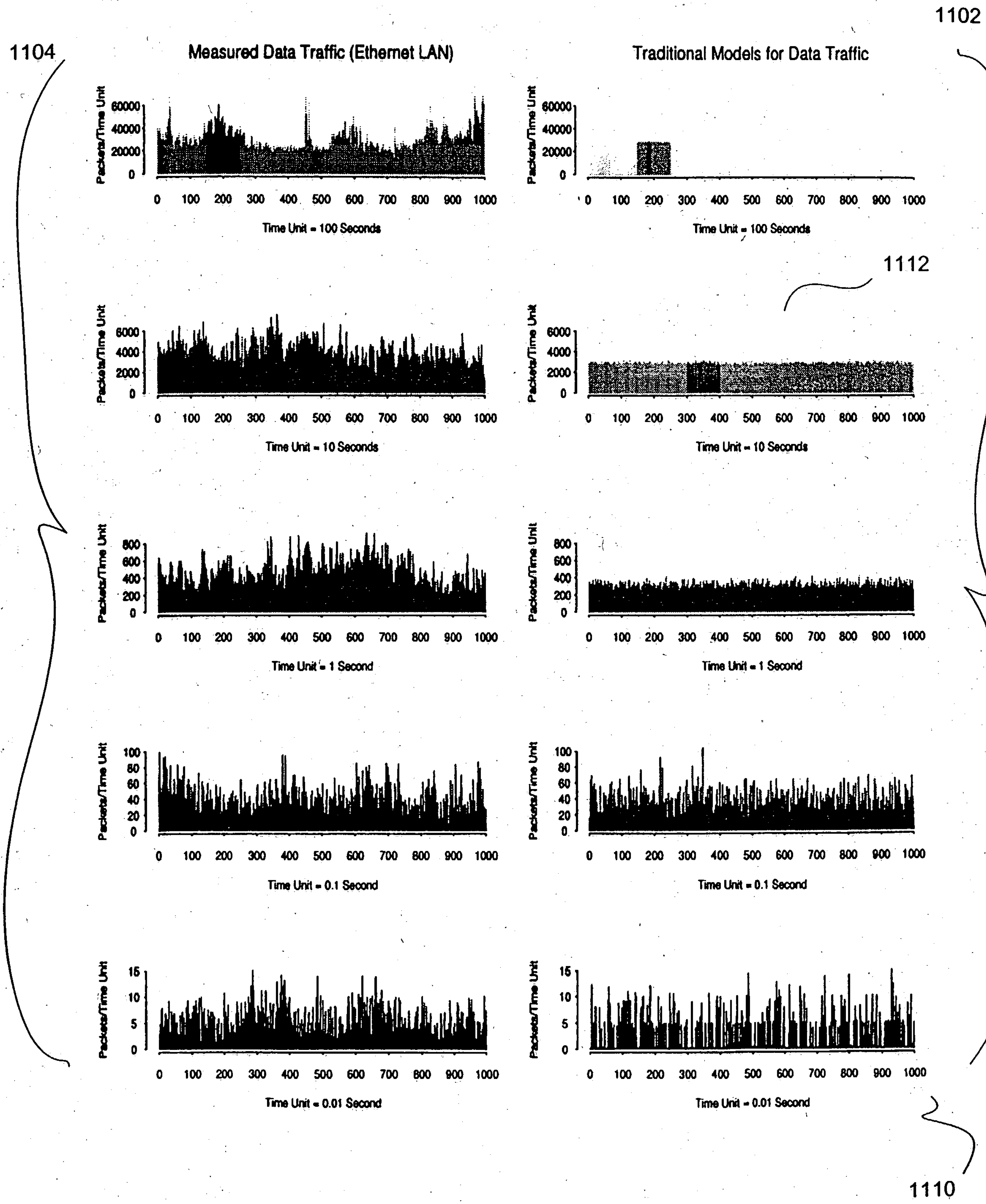


Figure 12

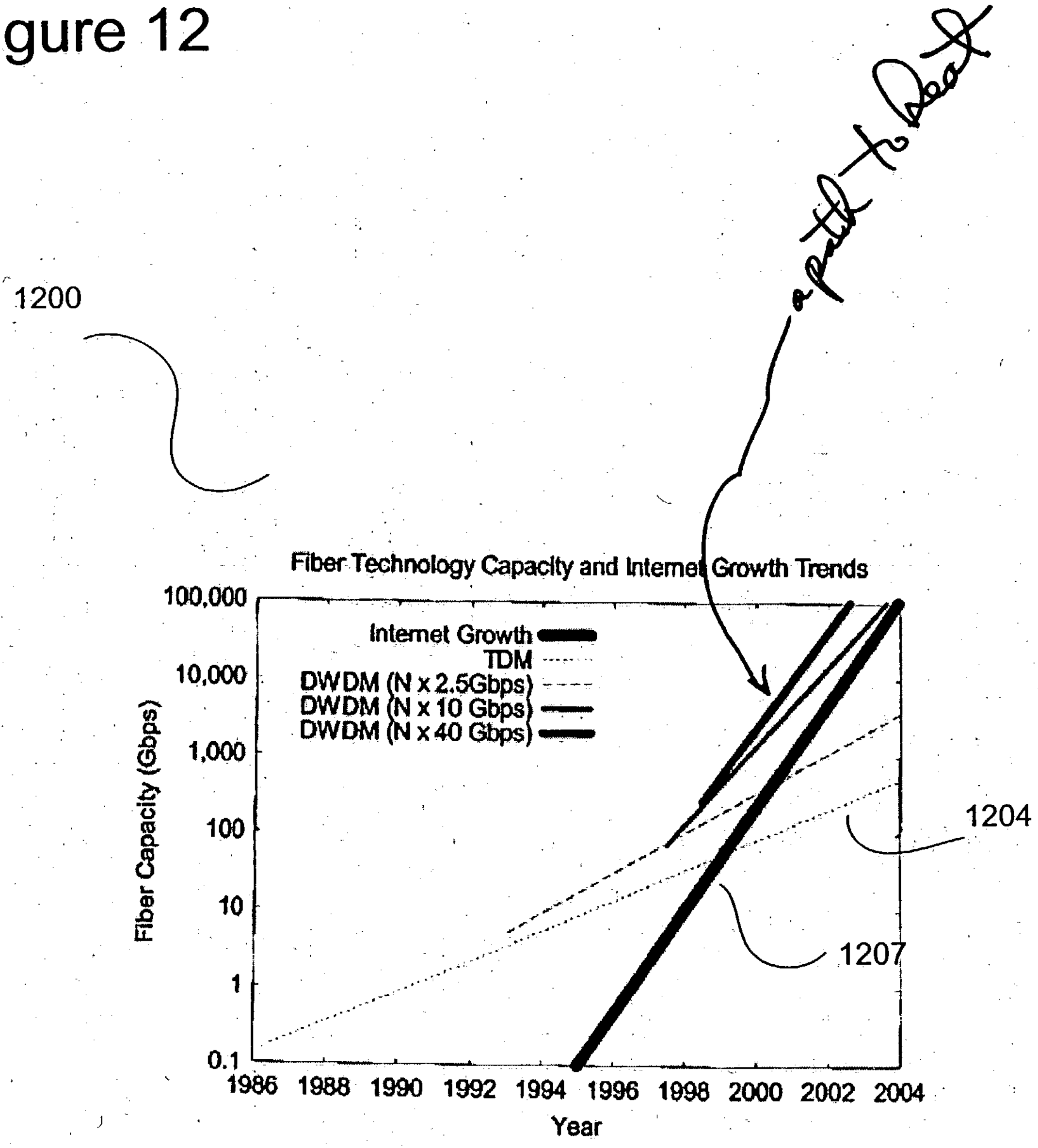

1200
a path to Deat
Fiber Technology Capacity and Internet Growth Trends
Internet Growth
TDM
DWDM (N x 2.5Gbps)
DWDM (N x 10 Gbps)
DWDM (N x 40 Gbps)
Fiber Capacity (Gbps)
100,000
10,000
1,000
100
10
1
0.1
1986
1988
1990
1992
1994
1996
1998
2000
2002
2004
Year
1204
1207

Figure 13

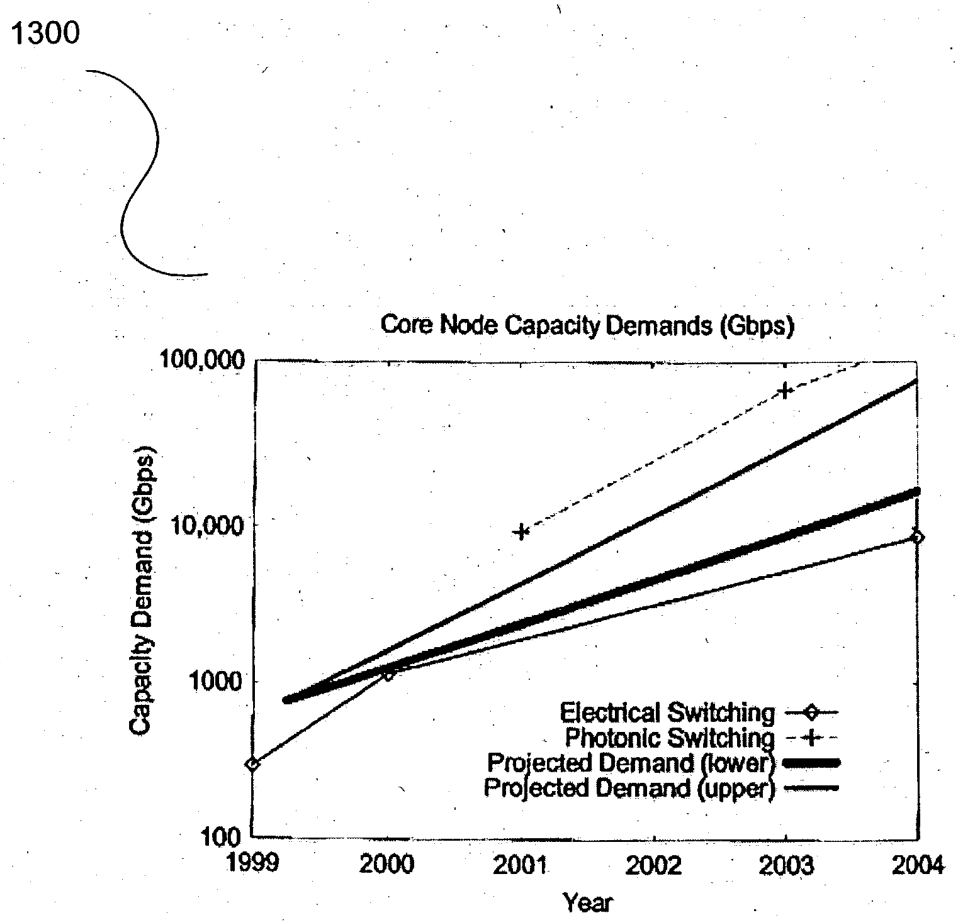
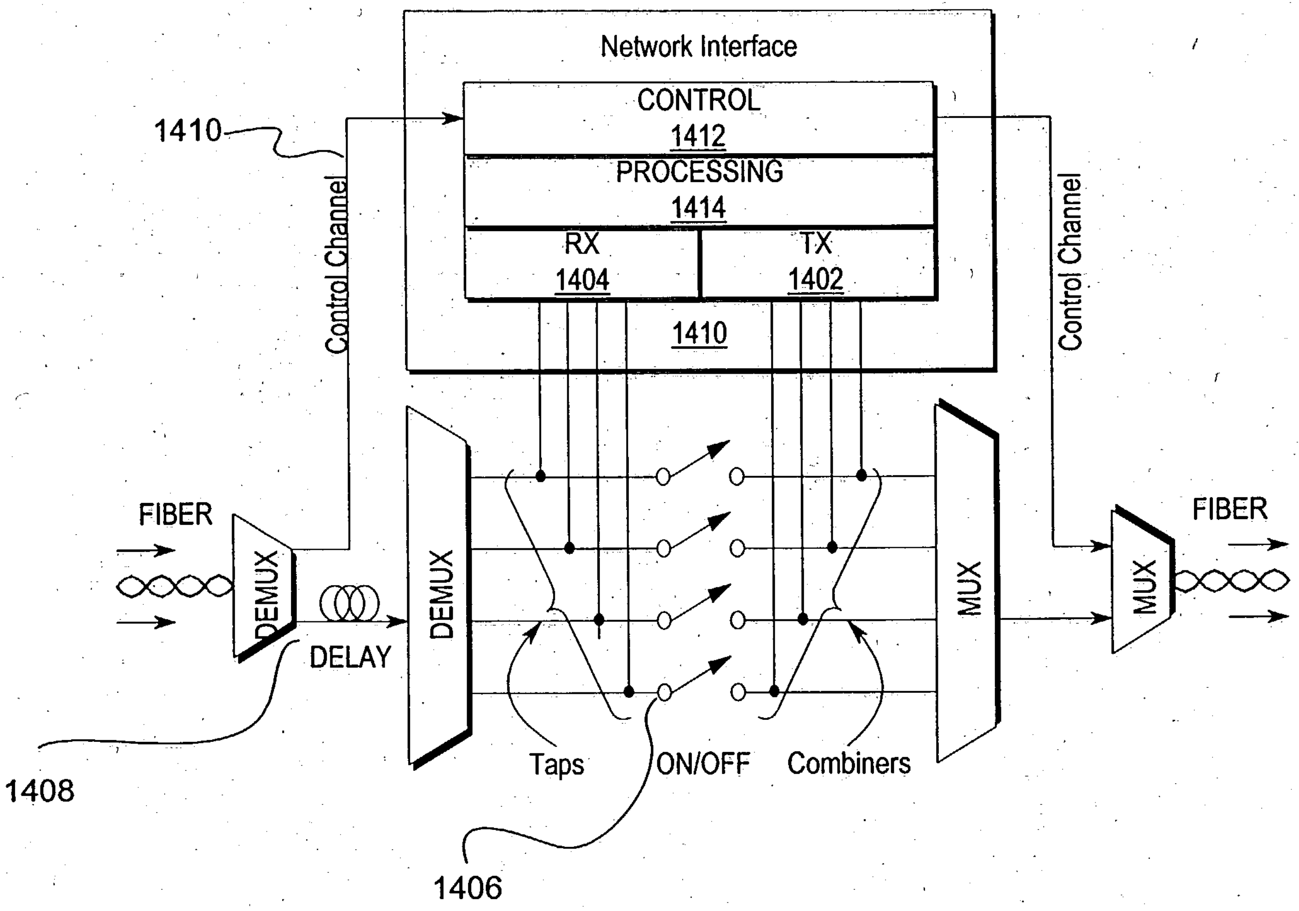

1300
Core Node Capacity Demands (Gbps)
100,000
10,000
1000
100
Capacity Demand (Gbps)
1999
2000
2001
2002
2003
2004
Year
Electrical Switching
Photonic Switching
Projected Demand (lower)
Projected Demand (upper)

Figure 14



PRIOR ART

Fumagalli/Cai/Chlamtac

[redacted]

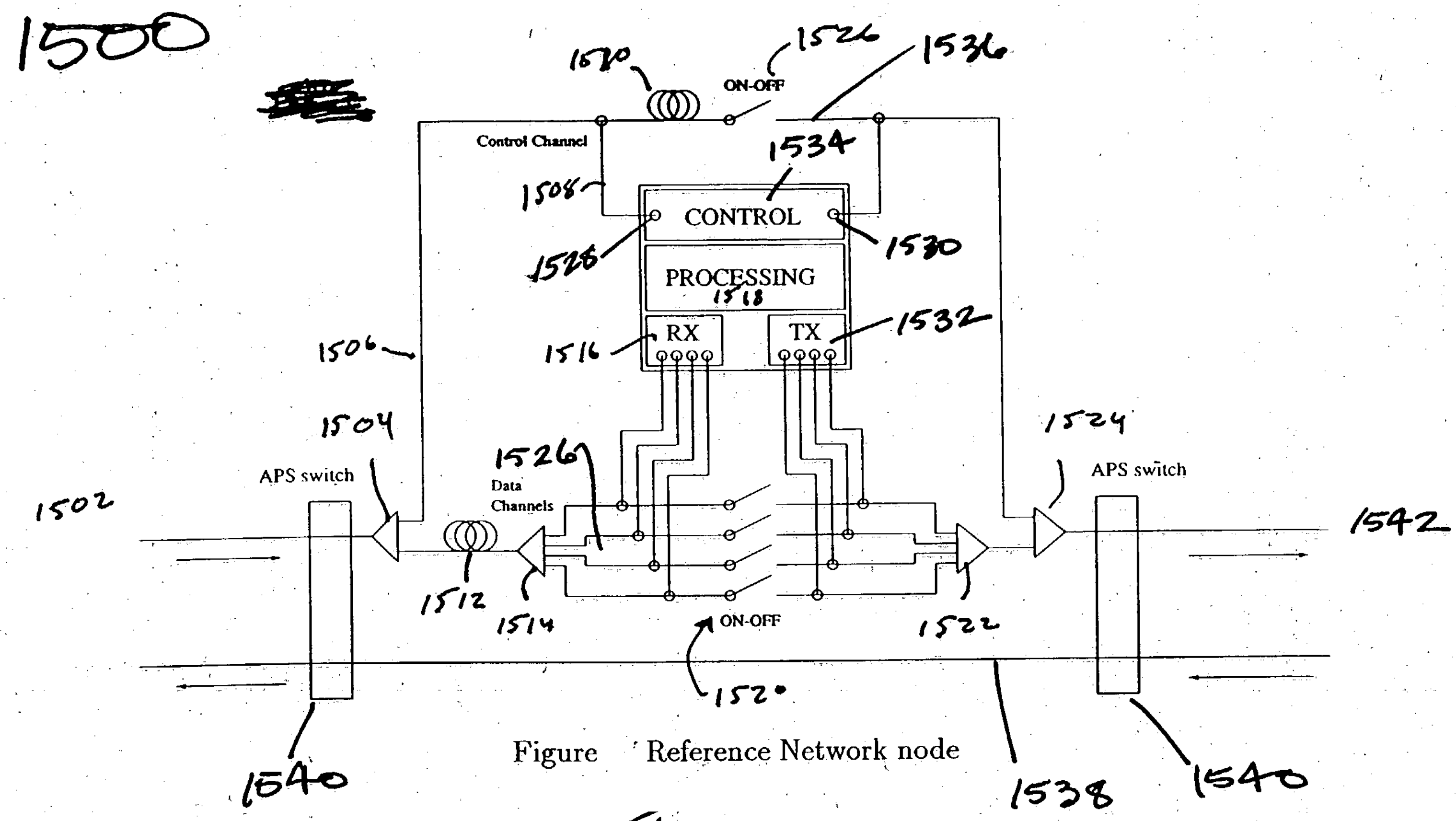
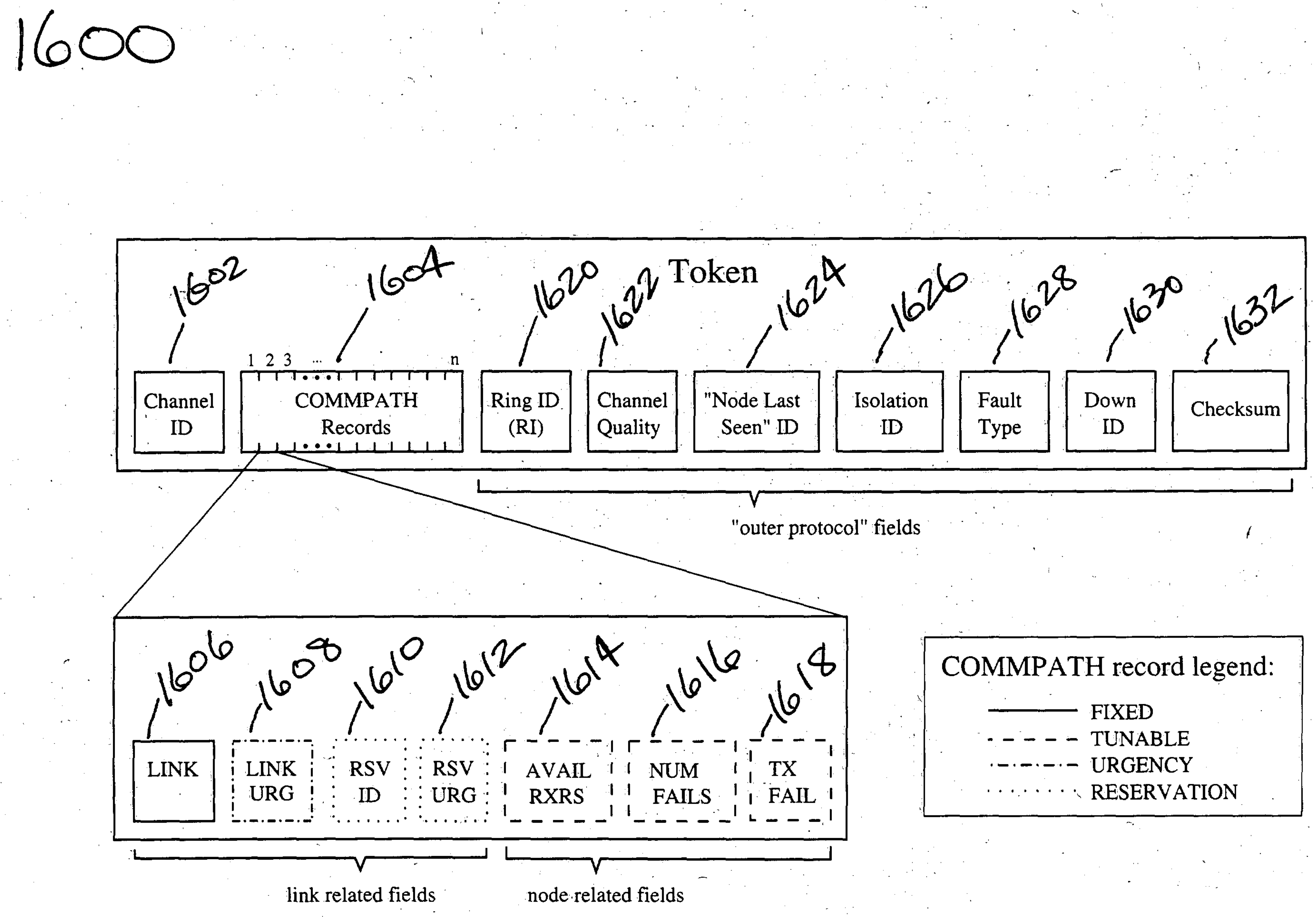


Figure 15: Reference Network node

f: neonode

patent app #1


1600
Token
1602
1604
1620
1622
1624
1626
1628
1630
1632
Channel ID
1 2 3 ... n
COMMPATH Records
Ring ID (RI)
Channel Quality
"Node Last Seen" ID
Isolation ID
Fault Type
Down ID
Checksum
"outer protocol" fields
1606
1608
1610
1612
1614
1616
1618
LINK
LINK URG
RSV ID
RSV URG
AVAIL RXRS
NUM FAILS
TX FAIL
link related fields
node related fields
COMMPATH record legend:
FIXED
TUNABLE
URGENCY
RESERVATION

1700

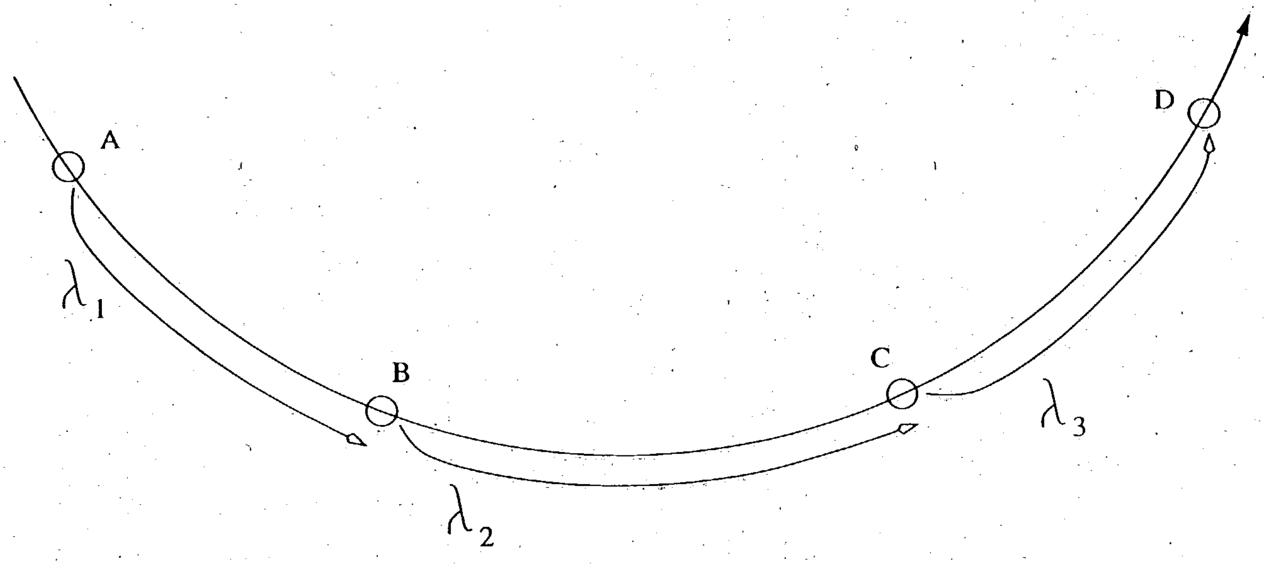


Figure Nodes A, B, and C contend for D's lone RXR

1.7:

1700

f: RXR-contention

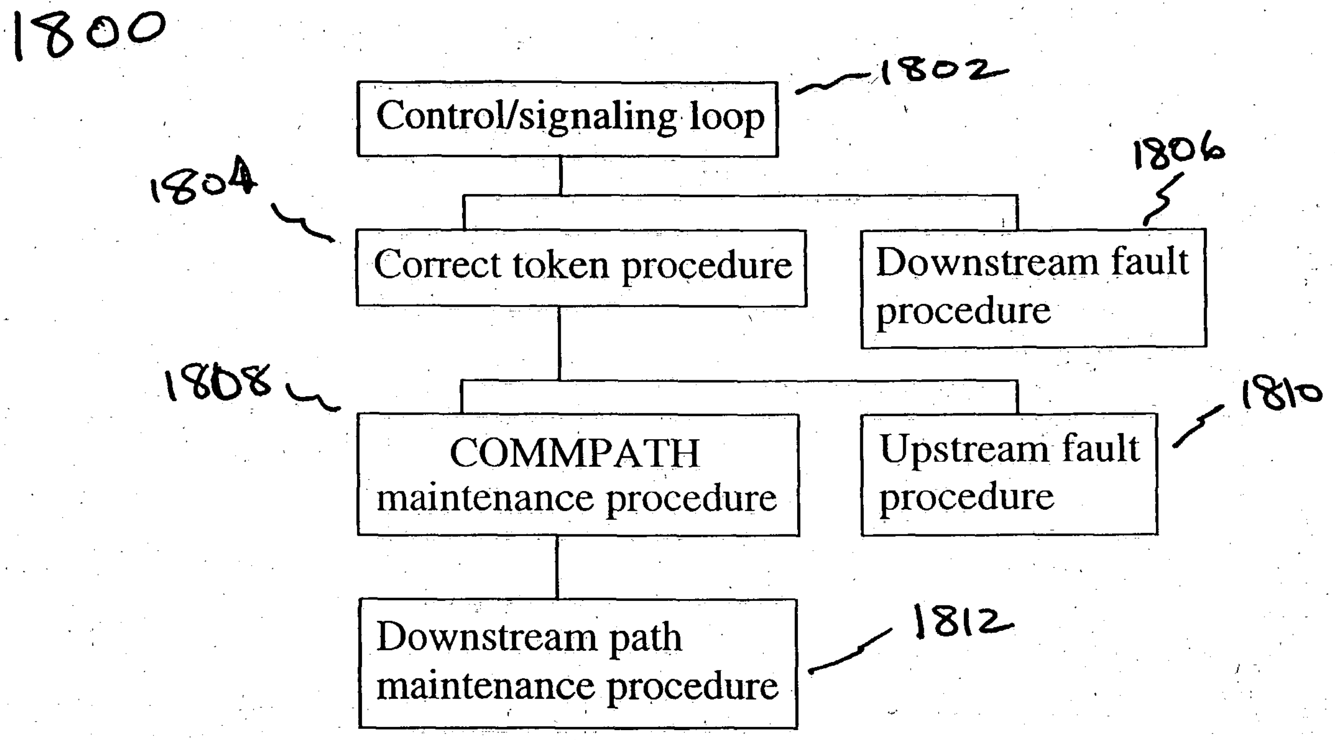
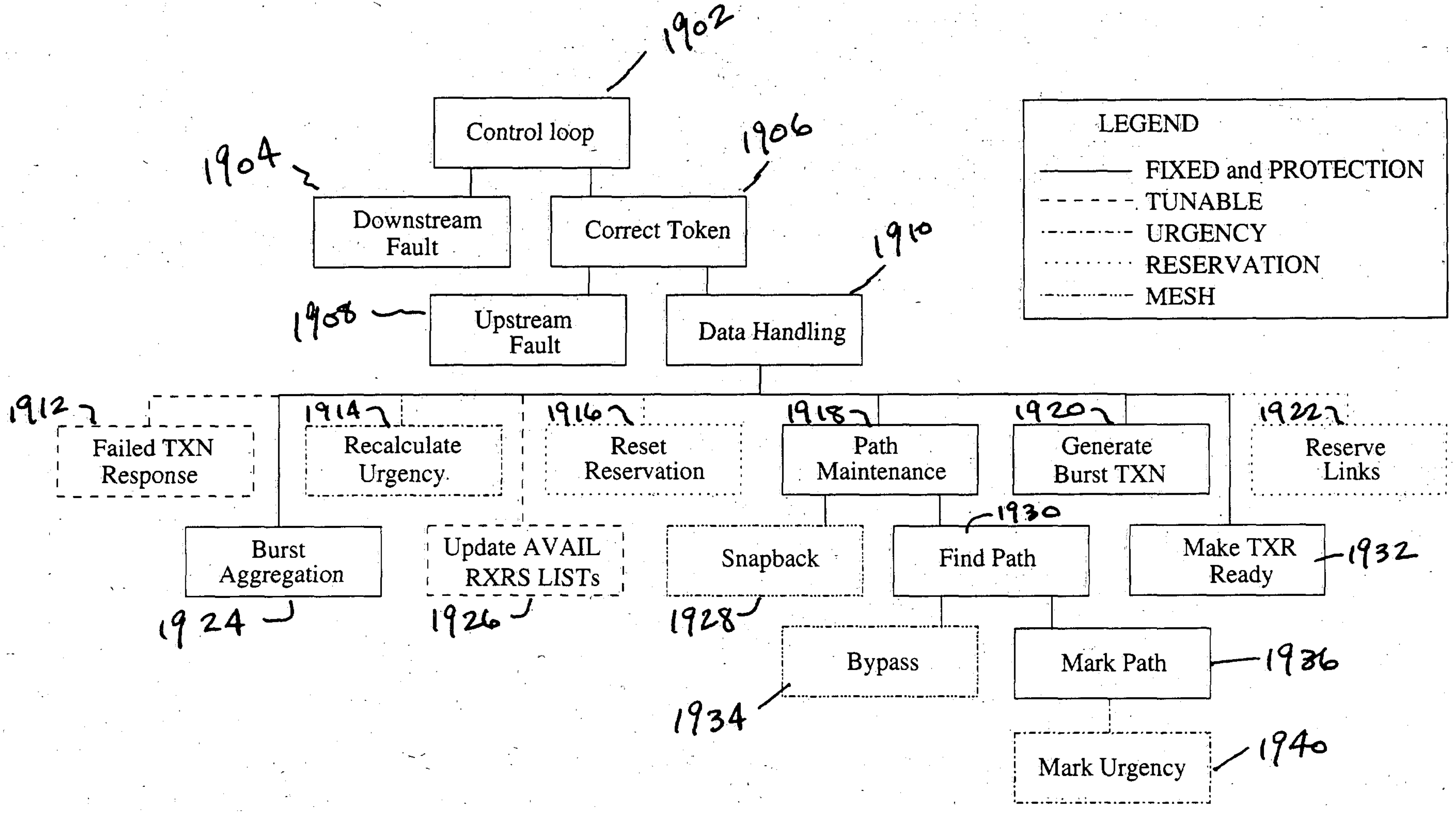


Figure Reference Network protocol procedure dependencies

18:

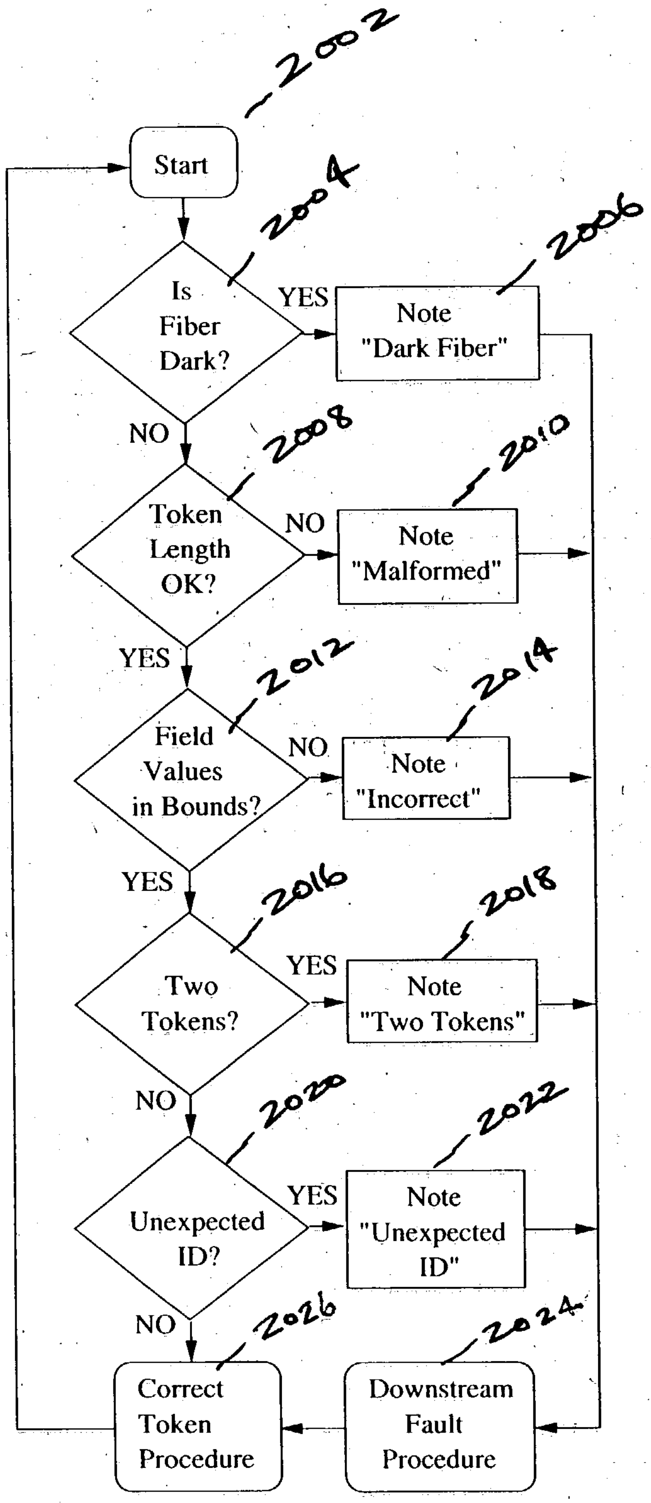
f:flow-ref-net-protocol-topview

1900



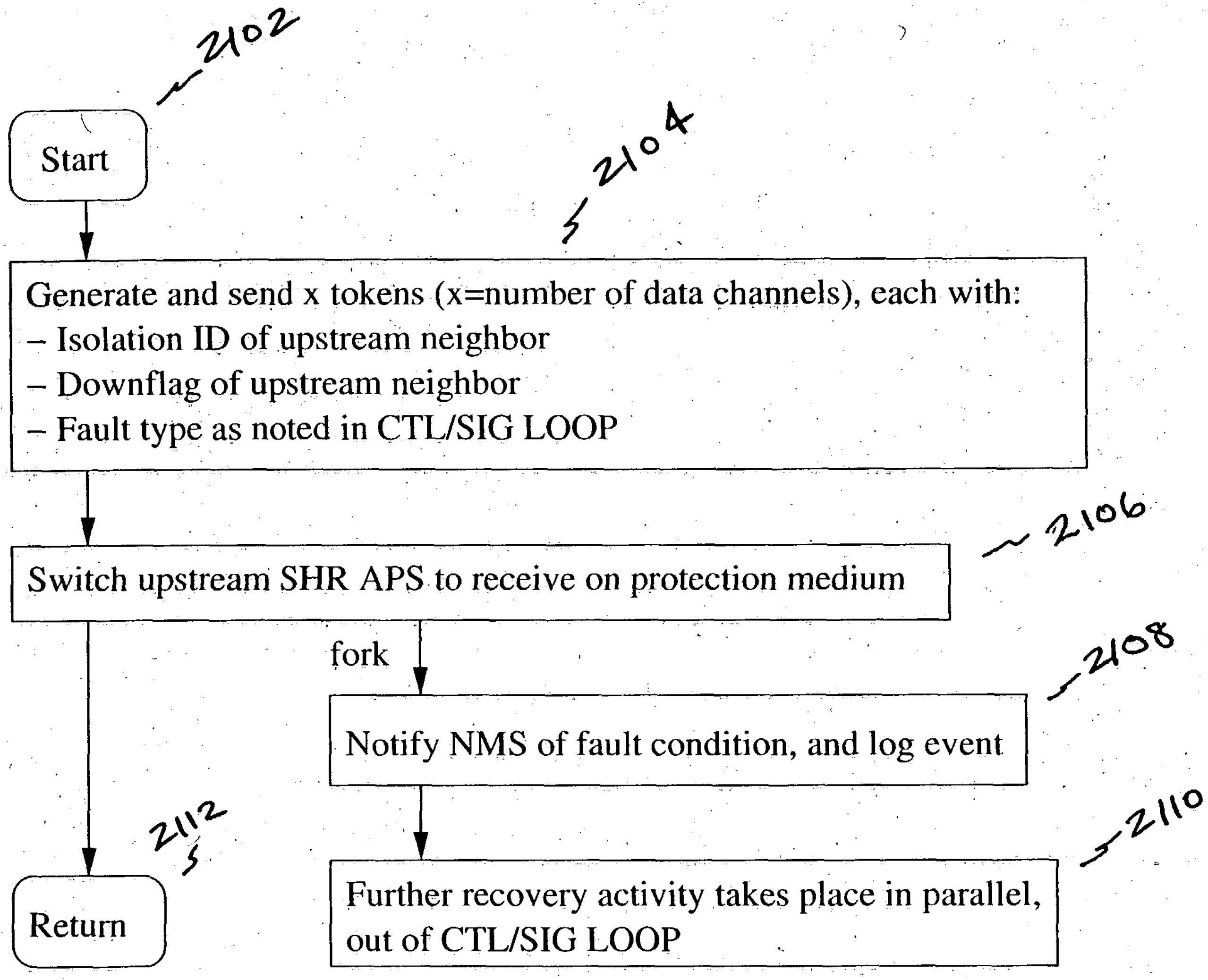
PROTOCOL PROCEDURE CALLING DEPENDENCIES

2000



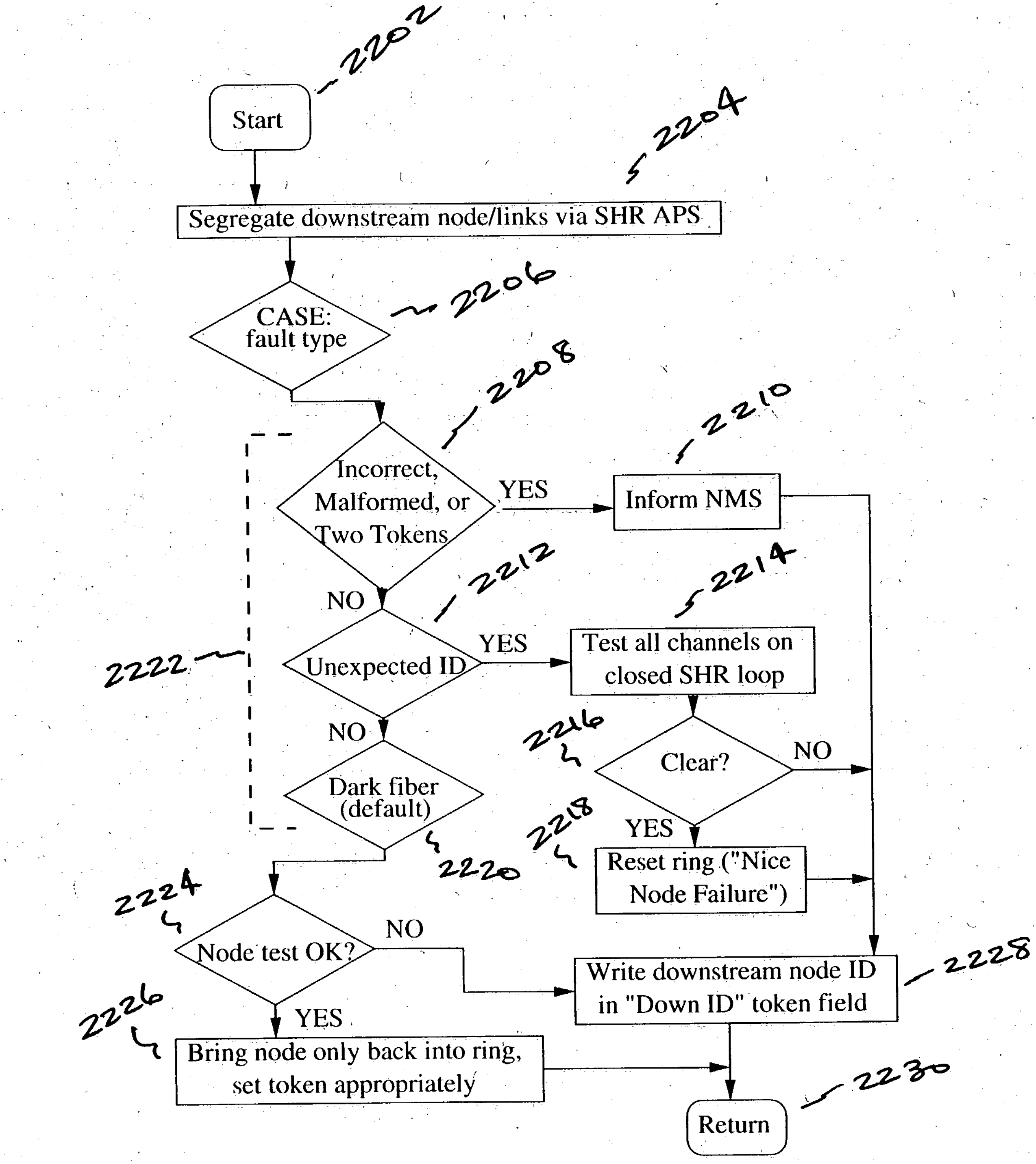
CONTROL/SIGNALING LOOP

2100


2102
Start
2104
Generate and send x tokens (x=number of data channels), each with:
– Isolation ID of upstream neighbor
– Downflag of upstream neighbor
– Fault type as noted in CTL/SIG LOOP
2106
Switch upstream SHR APS to receive on protection medium
fork
2108
Notify NMS of fault condition, and log event
2110
Further recovery activity takes place in parallel, out of CTL/SIG LOOP
2112
Return

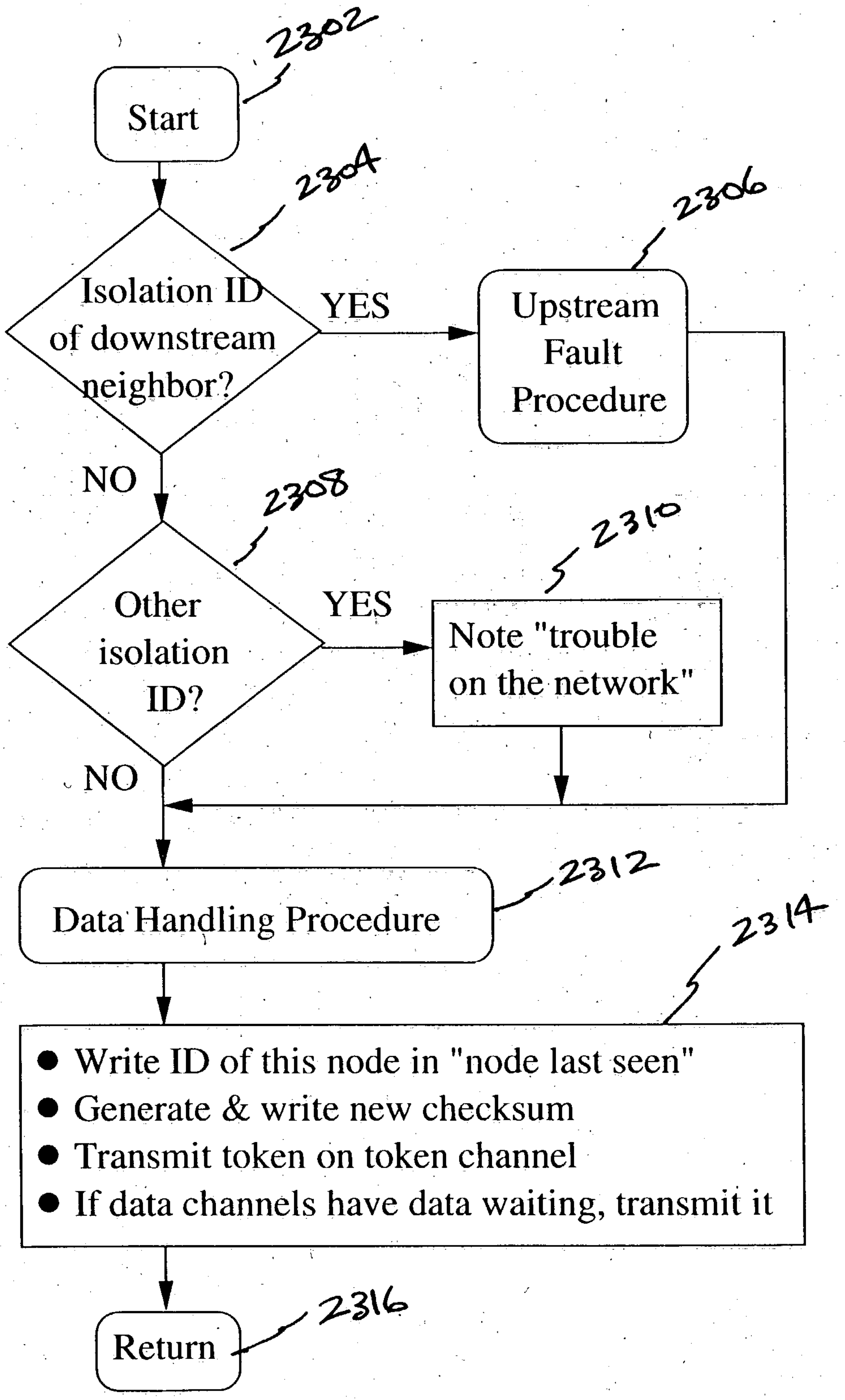
DOWNSTREAM FAULT PROCEDURE

2200

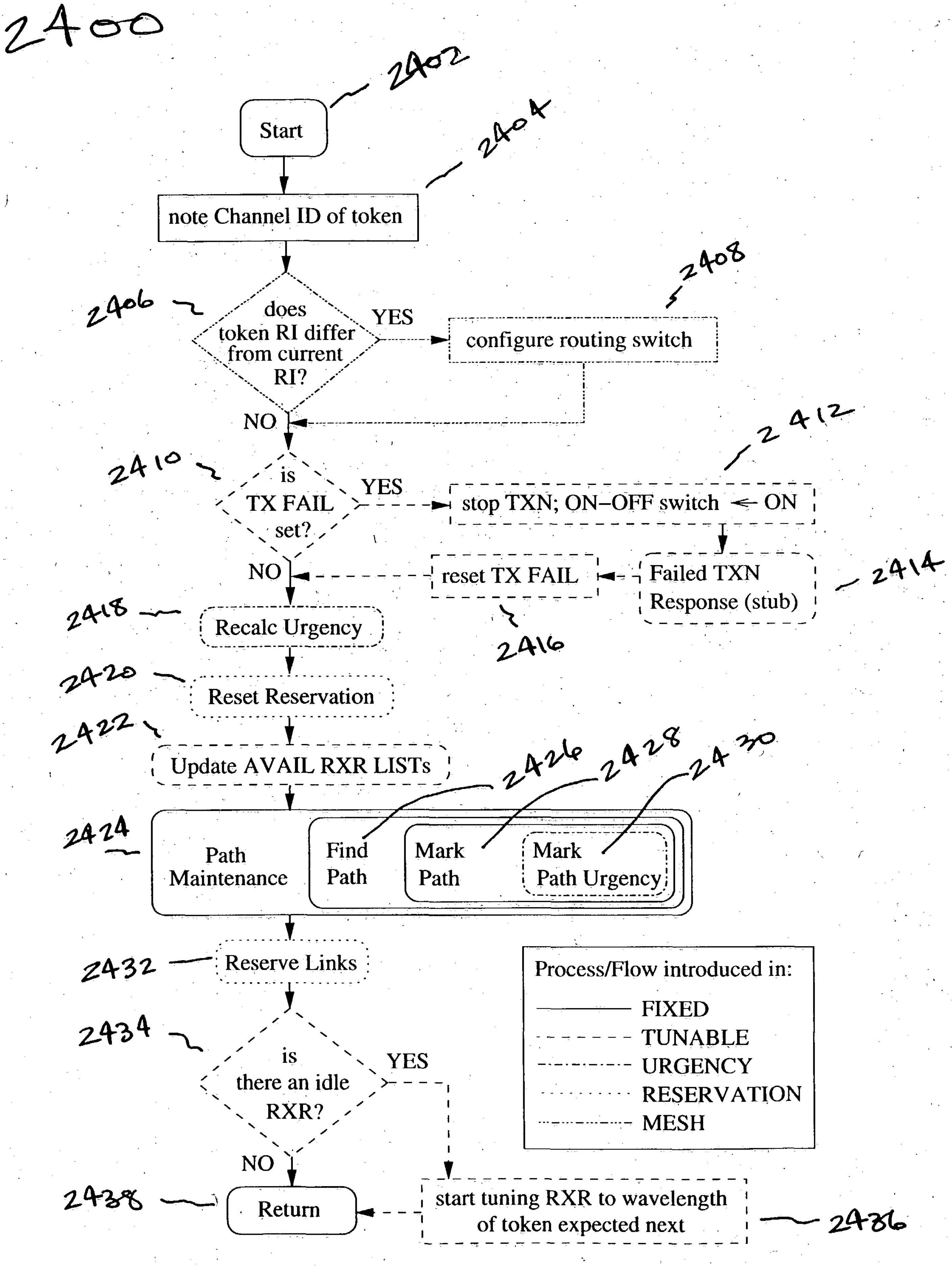

2202
Start
2204
Segregate downstream node/links via SHR APS
2206
CASE: fault type
2208
Incorrect, Malformed, or Two Tokens
YES
2210
Inform NMS
NO
2212
Unexpected ID
YES
2214
Test all channels on closed SHR loop
2222
NO
2216
Clear?
NO
YES
Dark fiber (default)
2220
2218
Reset ring ("Nice Node Failure")
2224
Node test OK?
NO
Write downstream node ID in "Down ID" token field
2228
2226
YES
Bring node only back into ring, set token appropriately
2230
Return

UPSTREAM ~~FAULT~~ PROCEDURE

2300

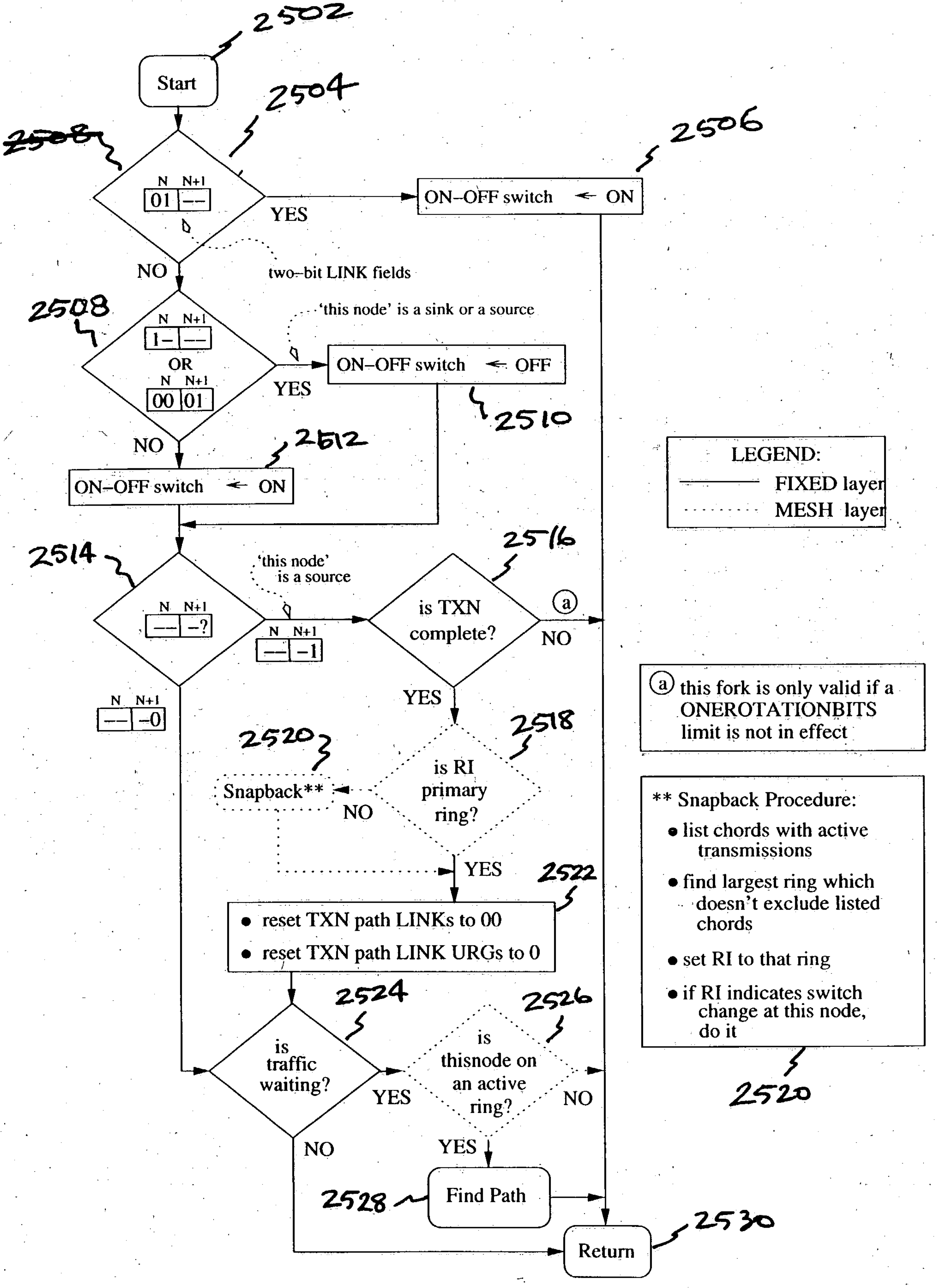

2302
Start
2304
Isolation ID of downstream neighbor?
YES
2306
Upstream Fault Procedure
NO
2308
Other isolation ID?
YES
2310
Note "trouble on the network"
NO
2312
Data Handling Procedure
2314
● Write ID of this node in "node last seen"
● Generate & write new checksum
● Transmit token on token channel
● If data channels have data waiting, transmit it
2316
Return

CORRECT TOKEN PROCEDURE



DATA HANDLING

2500



PATH MAINTENANCE

2600

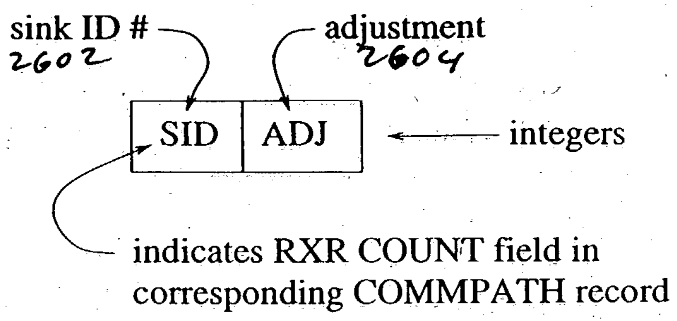


Figure RXR COUNT LIST record fields

26

f: flow2-rxr-count-list-record

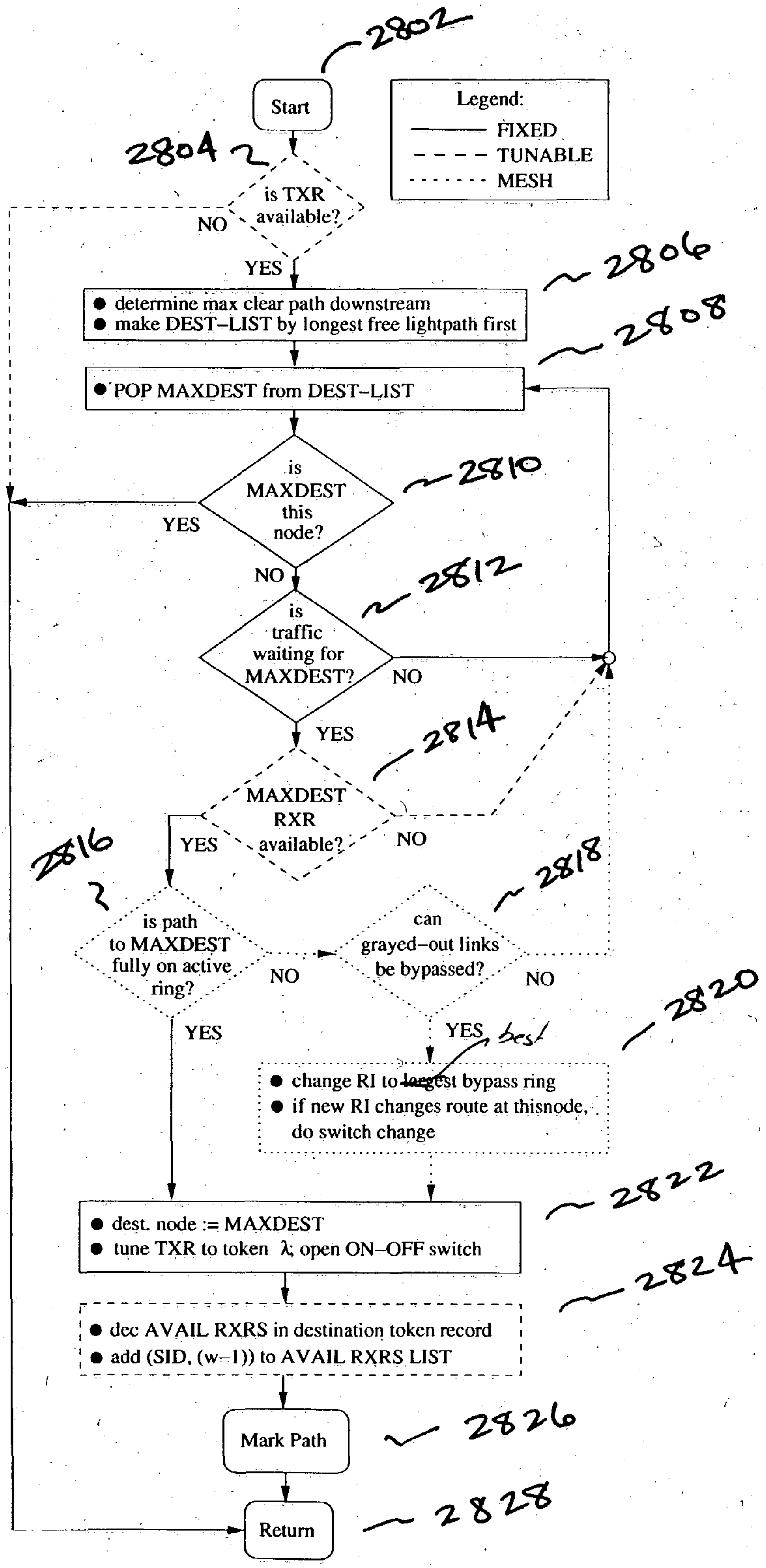
2700

```
Algorithm 0.0.2: UPDATE AVAIL RXR LISTS(Global, Node, Token)

if rxr_lists are empty                                                    [block 0]
  then return

if Node is on a lightpath                                                 [block 1]
  then note source and sink

for each $rec \in Node.add\_back\_rxr\_list$                             [block 2]
  do { increment rec.adj
       increment Token[rec.sink].AVAIL_RXRS
       if Token[rec.sink].NUM_FAILS $> 0$
         then decrement Token[rec.sink].NUM_FAILS
       if $rec.adj = 0$
         then delete rec

for each $rec_1 \in Node.take\_away\_rxr\_list$                          [block 3]
  do { decrement $rec_1.adj$
       decrement Token[$rec_1$.sink].AVAIL_RXRS
       if (Token[$rec_1$.sink].AVAIL_RXRS + Token[$rec_1$.sink].NUM_FAILS) $< 0$[a]
         then { increment Token[$rec_1$.sink].NUM_FAILS
                if sink noted and $rec_1.sink = sink$ and Token[$rec_1$.sink].LINK = SINK[b]
                  then if no active TXN to sink
                    then { comment: TANDEM
                           $Node.on\_off[\lambda_i] \leftarrow$ ON
                  else if Token[sink].LINK_URG $\geq$ urgency of least urgent active TXN
                    then { comment: STOMP
                           $Node.on\_off[\lambda_i] \leftarrow$ ON
                           discontinue own least urgent active TXN
                           invoke Failed_TXN()
                    else { comment: SIPHON
                           reset lightpath from sink upstream
                           $Node.on\_off[\lambda_i] \leftarrow$ OFF
                           Token[source].TX_FAIL $\leftarrow$ sink
       if $rec_1.adj = 0$
         then { new $rec_2 \leftarrow (rec_1.sink, -(Global.num\_tokens))$
                add $rec_2$ to Node.add_back_rxr_list
                delete $rec_1$
```

[a] if the sum of the AVAIL_RXRS and NUM_FAILS token fields for rec_1.sink becomes negative ...

[b] if rec_1.sink is the sink of a lightpath that was noted near the top of the algorithm ...

2800


2802
Start
Legend:
FIXED
TUNABLE
MESH
2804
is TXR available?
NO
YES
2806
● determine max clear path downstream
● make DEST-LIST by longest free lightpath first
2808
● POP MAXDEST from DEST-LIST
is MAXDEST this node?
2810
YES
NO
2812
is traffic waiting for MAXDEST?
NO
YES
2814
MAXDEST RXR available?
YES
NO
2816
is path to MAXDEST fully on active ring?
NO
2818
can grayed-out links be bypassed?
NO
YES
YES
best
2820
● change RI to ~~largest~~ bypass ring
● if new RI changes route at thisnode, do switch change
2822
● dest. node := MAXDEST
● tune TXR to token λ; open ON-OFF switch
2824
● dec AVAIL RXRS in destination token record
● add (SID, (w-1)) to AVAIL RXRS LIST
Mark Path
2826
Return
2828

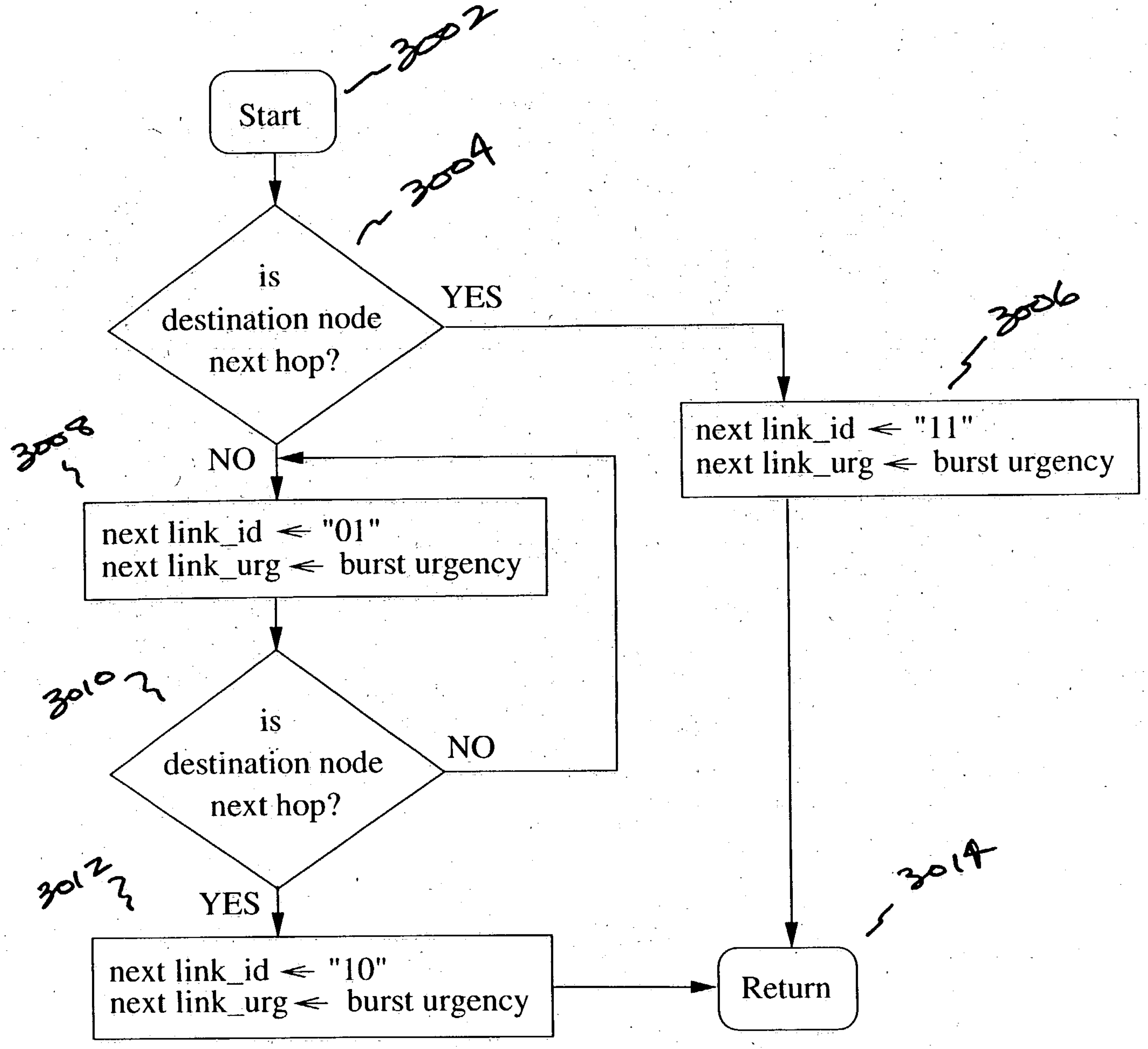
FIND PATH (FIXED. TUNABLE MESH)

2900

Algorithm 0.0.3: RESERVE LINKS PROCEDURE(*Token*, *Path*)

```
create priority queue of destination candidates, sorted on primary key (most urgent),
                                        and secondary key (greatest hop-length)
while (1)
    do  if queue empty
          then return
        pop candidate
        if every RSV_URG of path to candidate has lower urgency than candidate burst
          then drop through WHILE loop
for each link on Path
    do  note "losing" reservation ID, if any
        write "my" ID and MAXDEST urgency number
for each losing ID
    do  reset all reservations which are contiguous to new reservation link
              and which have losing IDs (clear away "orphaned" IDs)
```

3000


3002
Start
3004
is destination node next hop?
YES
3006
next link_id ← "11"
next link_urg ← burst urgency
3008
NO
next link_id ← "01"
next link_urg ← burst urgency
3010
is destination node next hop?
NO
3012
YES
next link_id ← "10"
next link_urg ← burst urgency
3014
Return

MARK PATH (MARK PATH URGENCY)

3100

Algorithm 0.0.1: FIND PATH(*Global*, *Node*, *Token*)

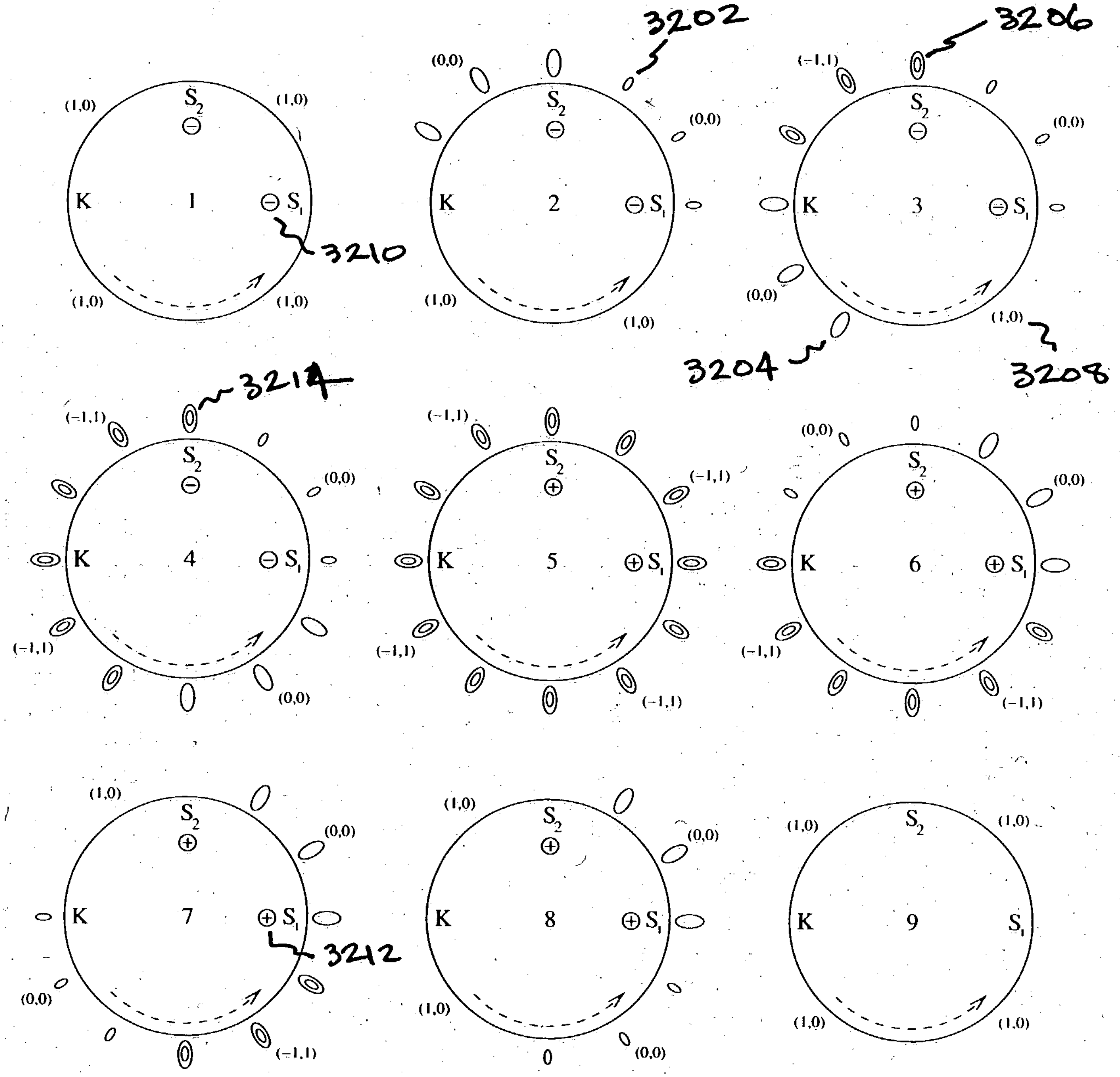
```
if a TXR is available
  then
    find $max$ (the FREE link farthest downstream on $active\ ring$)
    $dest\_list \leftarrow$ all dests (with bursts waiting) incl. $max$[a]
    sort $dest\_list$, by primary key (most urgent)                  [3110]
            and secondary key (farthest)
    while (1)
      do
        if $dest\_list$ is empty
          then return
        $dest \leftarrow$ pop $dest\_list$
        if ($\forall$ intermediate link, ($dest.urg > link.$RSV_URG)     [3118]
         and "grayed-out" links can be bypassed)                       [3120]
          then break[b]
    if "grayed-out" links on path                                      [3130]
      then
        $Token.RI \leftarrow$ largest available bypass ring (RI)
        if new RI changes the route at thisnode, set switch
    decrement $Token[dest]$.AVAIL_RXRS
    $arrec \leftarrow (dest, Global.num\_tokens - 1)$
    add $arrec$ to $Node.take\_away\_rxrs\_list$
    $mark\_path(dest, dest.urg)$
```

[a] Recall that node information appears on the token in the same record with its upstream link.

[b] The **break** statement is unconditional, except in RESERVATION_SCHEME (first condition) and MESH (second condition).

FIND PATH (URGENCY. RESERVATION)

3200



Phases of receiver accounting

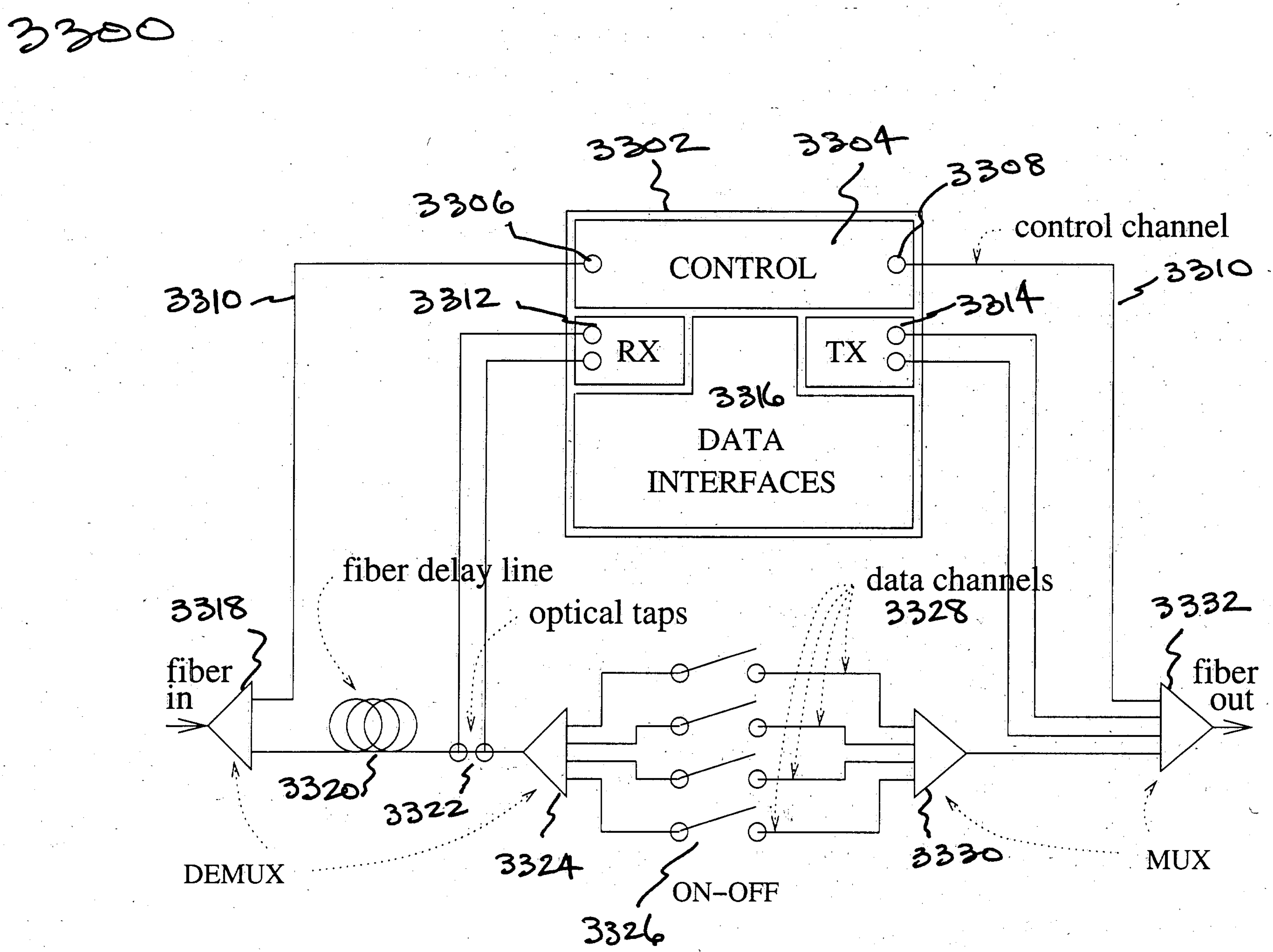


FIG. 33. A low-power mode, exclusive of protection hardware.

3400

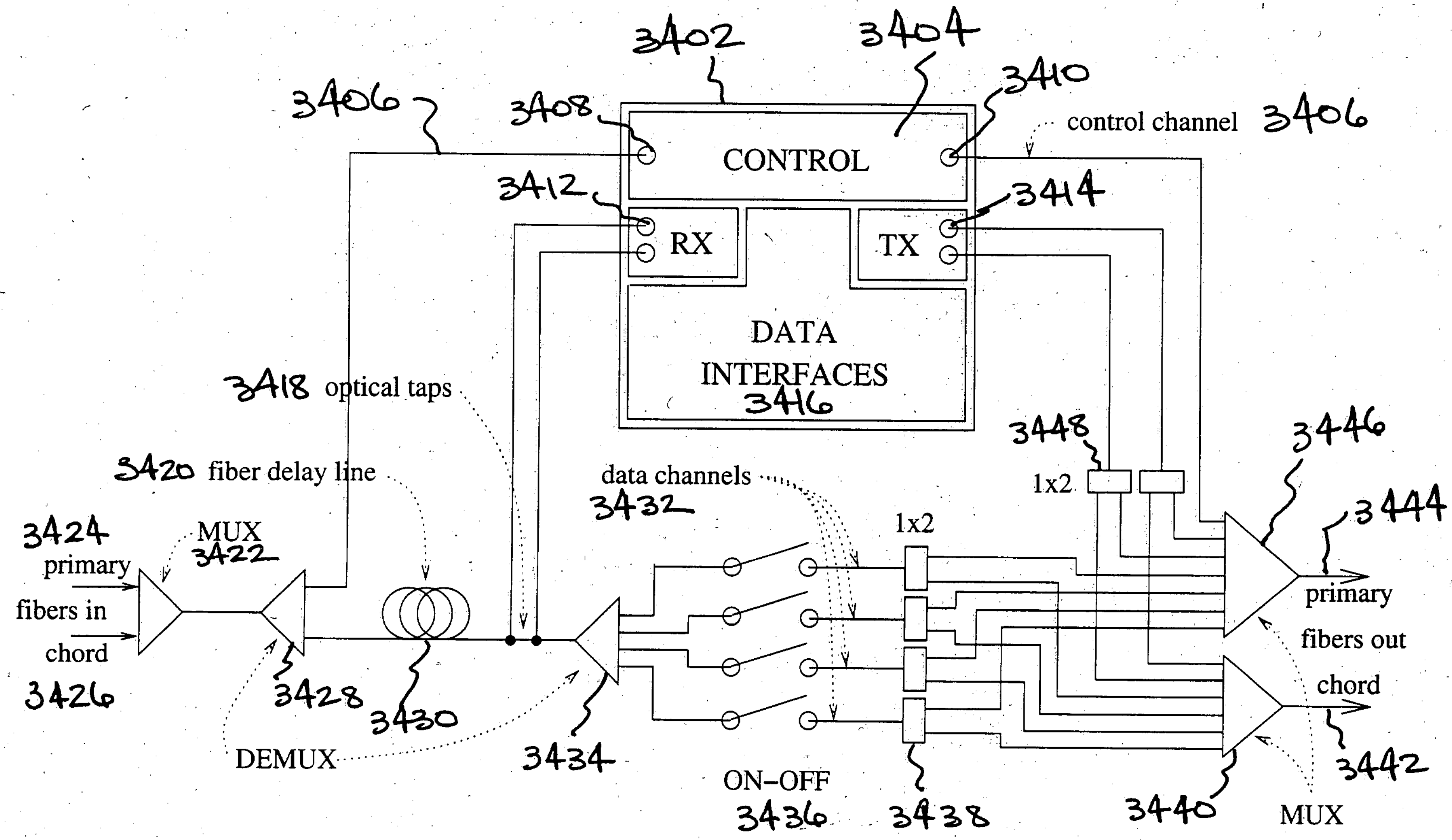


FIG. 34. A low-power node, in a MESH architecture.